U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

# SEARCH REQUEST FORM

Requestor's Name: Michael Richey  Serial Number: 08/187,662

Date: 3/2/95  Phone: 305-9669  Art Unit: 2316

**Search Topic:**

Please write a detailed statement of search topic. Describe specifically as possible the subject matter to be searched. Define any terms that may have a special meaning. Give examples or relevant citations, authors keywords, etc., if known. For sequences, please attach a copy of the sequence. You may include a copy of the broadest and/or most relevant claim(s).

Applicant's Nome: Robert A. ALFIERI
Filing date: 2/22/94
Claim 1 is attached

Other Keywords Include:
Threads, system calls
fast kernel trap or kernel functioncall

Basically, an operating system can perform system calls and kernel calls; however, it perfers to perform kernel calls. In some situations a system call can only be utilized to perform a certain event. Therefore, the operating system must interrupt kernel call processing and (promote or demote) or change to system call processing

**STAFF USE ONLY**

Date completed: 3/3/95
Searcher: DV
Terminal time: 60
Elapsed time: _____
CPU time: _____
Total time: 70
Number of Searches: 5
Number of Databases: 18

**Search Site**
- _____ STIC
- _____ CM-1
- 5 Pre-S

**Type of Search**
- _____ N.A. Sequence
- _____ A.A. Sequence
- _____ Structure
- _____ Bibliographic

**Vendors**
- _____ IG Suite
- _____ STN
- ✓ Dialog
- _____ APS
- _____ Geninfo
- _____ SDC
- _____ DARC/Questel
- _____ Other

PTO-1590 (9-90)

USCOMM-DC 90-3952
★U.S. GPO: 1994-300-608/00105

SYSTEM:OS - DIALOG OneSearch
File 350:Derwent World Pat. 1963-1980/UD=9504
(c) 1995 Derwent Info Ltd
File 351:DERWENT WPI 1981-1995/UD=9507;UA=9503;UM=9444
(c)1995 Derwent Info Ltd
*File 351: Free images in March and April. Also, help celebrate WPI's 7 millionth record. Enter HELP NEWS 351 for more information.

Set Items Description

ds

| Set | Items | Description |
|---|---|---|
| S1 | 1011 | KERNEL |
| S2 | 822734 | SYSTEM |
| S3 | 36031 | CALL? ? OR CALLED OR CALLING |
| S4 | 24 | S1 AND S2 AND S3 |
| S5 | 47937 | INTERRUPT? |
| S6 | 2 | S4 AND S5 |

?t 6/7/1-2

6/7/1 (Item 1 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1995 Derwent Info Ltd. All rts. reserv.

009580838 WPI Acc No: 93-274384/35
XRPX Acc No: N93-210708
Computer system with demand loading of data segment register - defines page table used to prevent application program from accessing data segment so that it can be shared by system routine
Patent Assignee: (MICR-) MICROSOFT CORP
Author (Inventor): WILLMAN B M
Number of Patents: 002
Number of Countries: 005
Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| EP | 557908 | A2 | 930901 | 9335 | (Basic) |
| CA | 2090194 | A | 930827 | 9346 | |

Priority Data (CC No Date): US 843994 (920226)
Applications (CC,No,Date): CA 2090194 (930223); EP 93102677 (930219)
Language: English
EP and/or WO Cited Patents: No-SR.Pub
Designated States
(Regional): DE; FR; GB; IT
Abstract (Basic): EP 557908 A

Operating system routines are loaded into pages of a page table that are only accessible in kernel modes. An application program is loaded into pages that are accessible in both user and kernel modes. The application program is executed in user mode and control is transferred to an operating system routine where the computer switches to kernel mode.

While executing the operating system , control is transferred to an instruction that uses the data segment register and an exception is generated when the data segment register contains a selector other than the selector for the defined data segment. The execution handler is executed and control transferred to the appropriate instruction.

ADVANTAGE - Minimises the number of segment register loads that occur during system calls and interrupt processing. Avoids

segment register loads during process or thread switch time. Provides secure environment for each task

Dwg.5/7

Derwent Class: T01;
Int Pat Class: G06F-009/44; G06F-009/445; G06F-012/02; G06F-012/10

6/7/2 (Item 2 from file: 351)
DIALOG(R)File 351:DERWENT WPI


009281449 WPI Acc No: 92-408860/50
XRPX Acc No: N92-311835 *Image available*

Diagnostic system for personal computer - has error log for storing set error log information at set locations and information is accessible by various programs

Patent Assignee: (IBMC ) INT BUSINESS MACHINES CORP
Author (Inventor): TREU A R
Number of Patents: 002
Number of Countries: 004
Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| EP | 517403 | A2 | 921209 | 9250 | (Basic) |
| US | 5245615 | A | 930914 | 9338 | |

Priority Data (CC No Date): US 711003 (910606)
Applications (CC,No,Date): EP 92304650 (920521)
Language: English
EP and/or WO Cited Patents: No-SR.Pub
Designated States (Regional): DE; FR; GB

Abstract (Basic): EP 517403 A

The diagnostic system is described for a personal computer which has a storage system for a number of programs including application programs, an operating system and a B10S. The B10S contains a number of routines which are functionally layered beneath the operating system and are independent but are accessible from the operating system by an interface of a number of B10S interrupt calls.

The diagnostic system has a non-volatile memory having a first address space for storing an error log, the error log comprising a number of set addressable locations for storing set error information in a set format. The B10S includes a number of diagnostic related routines including a first routine for writing error information into the error log and a second routine for reading error information from the error log.

ADVANTAGE - Operable under different operating systems.

Dwg.1/8

Abstract (US): 9338 US 5245615 A

A personal computer has a NVRAM comprising an error log for storing predetermined error log information at predetermined locations therein. The information is accessible by various programs such as a POST program, a diagnostics program, and an operating system program. Access is made by BIOS interrupt calls through a BIOS interface. The NVRAM also stores vital product data and system setup data.

After the hardware and software have been installed, computer is restored in step (104) by turning power back on and resetting the system by pressing simultaneously control, alt, and delete keys of keyboard. Upon successful completion of POST (106), OS kernel is

loaded in step (108) into memory and OS is initialised in step (116) where application programs are executed or run under control of OS. Errors are most frequent during system operation to step (116).

ADVANTAGE - Provides improved diagnostic system and interface for open ended PC that is operable under different operating systems, or in basic I/O operating system for logging and accessing error information in a non-volatile random access memory (NVRAM).

Dwg.2/8

Derwent Class: T01;
Int Pat Class: G01R-031/28; G06F-011/22

```
?s switch?
      S7  392494  SWITCH?
?ds
ds

Set     Items   Description
S1       1011   KERNEL
S2     822734   SYSTEM
S3      36031   CALL? ? OR CALLED OR CALLING
S4         24   S1 AND S2 AND S3
S5      47937   INTERRUPT?
S6          2   S4 AND S5
S7     392494   SWITCH?
S8          4   S4 AND S7
S9      44200   FAULT? OR COMPLICATION?
S10         2   S4 AND S9
S11         6   S8 OR S10
S12         5   S11 NOT S6
?t 12/7/1-5

 12/7/1      (Item 1 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1995 Derwent Info Ltd. All rts. reserv.
```

009755404 WPI Acc No: 94-035255/04
XRPX Acc No: N94-027386 *Image available*

Run time binding of software for computer system - uses trader which dynamically directs execution processes to either new or old versions of software depending upon creation times

Patent Assignee: (TELF ) TELEFONAKTIEBOLAGET ERICSSON L M
Author (Inventor): LUNDIN L K; MARKSTROM U K H; LUNDIN K; MARKSTROEM U; MARKSTROEM U K H
Number of Patents: 003
Number of Countries: 022
Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| WO | 9401818 | A1 | 940120 | 9404 | (Basic) |
| AU | 9345163 | A | 940131 | 9422 | |
| US | 5339430 | A | 940816 | 9432 | |

Priority Data (CC No Date): US 907307 (920701)
Applications (CC,No,Date): WO 93SE416 (930511); AU 9345163 (930511)
Language: English
EP and/or WO Cited Patents: 02Jnl.Ref; EP 518195; GB 2242293; GB 2258068; JP 1239633; JP 1307825; US 5093916; US 5175828
Designated States
(National): AU; BR; FI; KR; NO
(Regional): AT; BE; CH; DE; DK; ES; FR; GB; GR; IE; IT; LU; MC; NL; PT; SE
Filing Details: AU9345163 Based on WO 9401818
Abstract (Basic): WO 9401818 A

The system for dynamic run-time binding of software modules, includes a trader unit controlling linking. The trader unit (80) resides with the system kernel (82). When calls are made from a software unit (100), they can be directed to either a new (104) or the old (102) version of software being replaced. When making the replacement, the server classes from both old and new units have their interfaces 'published' in the trader. The trader contains address entries for both old and new units.

When replacement is complete, the old version is removed. An exception function is provided to coerce removal.

ADVANTAGE - Allows replacement of software units without loss of system operation.

Dwg.6/8

Abstract (US): 9432 US 5339430 A

Software is frequently modified, enhanced or replaced altogether by new versions. The implementation or integration of the new or revised software into the operational system must be accomplished in accordance with strict requirements for not disturbing the ongoing activities of the system . Therefore, it is desirable that the system not be halted while the change to the new software is made.

the preferred approach is to be able to replace software modules with new versions on the fly, during system operation. The smooth modification made possible in the disclosed system allows such changes with minimal disturbance to ongoing activities by dynamically linking and binding software modules during execution. The system accomplishes this by applying expanded object-oriented programming techniques and utilizing language-independent interface specifications that remain unchanged and that obviate the need for storing symbolic information that would be subject to change following modification.

USE/ADVANTAGE - In telecommunications switching systems. Efficiently performs dynamic runtime linking between separately loaded program units in computer system ,.

Dwg.5/8

Derwent Class: T01;

Int Pat Class: G06F-009/44

12/7/2 (Item 2 from file: 351)

DIALOG(R)File 351:DERWENT WPI



009479771 WPI Acc No: 93-173306/21

XRPX Acc No: N93-132871

Providing data protection between trusted and untrusted code segments in single process model - using system services that grant and remove addressability to sensitive data.

Patent Assignee: (ANON ) ANONYMOUS

Number of Patents: 001

Number of Countries: 001

Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| RD | 348003 | A | 930410 | 9321 | (Basic) |

Priority Data (CC No Date): RD 93348003 (930320)

Abstract (Basic): RD 348003 A

A text address range is registered as trusted rather than an entire process as trusted. System services can be provided that grant and remove addressability to sensitive data. The system services that grant addressability verify that control is being returned to a trusted section of code. The trusted section of code

always removed addressability before returning to the untrusted code section. The trusted code section could exist as are built subroutine library.

The concept is similar to user/ kernel boundary that already exists. When running user text, a process has restricted access to data. A system call switches to kernel text rather than user text and also grants access to kernel dta. The protection is provided by guaranteeing that control transfers to a trusted text segment.

ADVANTAGE - Permits the trusted and non-trusted code segments to run alternately on same process thread avoiding communication and process switching overhead.

Dwg.0/0

Derwent Class: T01;
Int Pat Class: G06F-000/00

12/7/3 (Item 3 from file: 351)
DIALOG(R)File 351:DERWENT WPI


008330995 WPI Acc No: 90-217996/29
XRPX Acc No: N90-169161 *Image available*

Real-time Fourier transformation system - uses network of butterfly circuits that are programmed for high rates of data handling; RADAR SIGNAL PROCESS COORDINATE

Patent Assignee: (DIEH ) DIEHL GMBH & CO
Author (Inventor): GEISSLINGE K; MUELLER H; WERGEN G; GEISSLINGER K; MULLER H
Number of Patents: 006
Number of Countries: 004
Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| DE | 3900349 | A | 900712 | 9029 | (Basic) |
| FR | 2641631 | A | 900713 | 9035 | |
| GB | 2229299 | A | 900919 | 9038 | |
| DE | 3900349 | C | 901011 | 9041 | |
| US | 5028877 | A | 910702 | 9129 | |
| GB | 2229299 | B | 930113 | 9302 | |

Priority Data (CC No Date): DE 3900349 (890107)
Applications (CC,No,Date): GB 90157 (900104); GB 90157 (900104); US 454514 (891221)
Filing Details: US5028877 (1810RMC)
Abstract (Basic): DE 3900349

A complex time signal (21) from such as a radar system is subjected to a fast Fourier-transformation to obtain frequency data. The signal is quantised (23) and passed through a weighting filter (25) coupled to a bit inverting address converter (26).

The values are entered into a series to parallel converter (27) and a multiplexer (29) loads the first half into a RAM buffer (28). A further multiplexer allows pairs of works to be received by a programmed controller (30) that has so called butterfly circuits (31). Each circuit has a complex variable multiplier to generate the transformation coefficients.

ADVANTAGE - Improved Fourier transformation speed. @(4pp Dwg.No.1/2

Abstract (US): 9129 US 5028877

The circuit arrangement implements fast discrete Fourier transform in real time through the controlled operation of cross-linked

butterfly, or kernel , operators. The circuit successively transmits two halves of a sequence of complex input words through a series-parallel input register and an interim data storage to a number of butterfly operators which operate in parallel. The outputs are switchable by a multiplexer for recursive linkage with the interim storage or, in essence, for the delivery of the frequency range-output words to a parallel-series output register. @(4pp)@

Abstract (GB): 9302 GB 2229299 B

A circuit arrangement for real-time performance of fast discrete Fourier transformation by controlled operation of cross-linked butterfly operators, characterised in that successively the two halves of a sequence of complex input words are transferred by way of a series/parallel input register and an intermediate store to a plurality of butterfly operators working in parallel, the outputs of which operators are switchable for recursive transfer to the intermediate store or for the issuance of frequency-range output words to a parallel/series output register.

Dwg. 1,2

Abstract (DE): 9041 DE 3900349

The circuit arrangement is based on the controlled operation of four interlaced butterfly operators. One after the other, both halves of a sequence of complex input words are passed to the butterfly operators working in parallel through a series-parallel input register and an intermediate memory.

The outputs of the operators are switched by a multiplexer either for a recursive combination process to the intermediate memory or for the release of the frequency-rnage output words to a parallel-series output register.

USE/ADVANTAGE - For radar equipment. Improvement in throughput and increase in speed.@(4pp

Derwent Class: T01; W06; R27; R19

Int Pat Class: G01S-013/00; G06F-015/33; G06F-015/332

12/7/4 (Item 4 from file: 351)

DIALOG(R)File 351:DERWENT WPI



007203140 WPI Acc No: 87-200149/29

XRPX Acc No: N87-149823

I-O accessing control in multi-tasking virtual memory data processor has memory manager controlling transfer of information between primary and secondary storage devices in response to page fault occurrence

Patent Assignee: (IBMC ) IBM CORP; (IBMC ) INT BUSINESS MACHINES CORP

Author (Inventor): DUVALL K E; HOOTEN A D

Number of Patents: 006

Number of Countries: 016

Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| EP | 229691 | A | 870722 | 8729 | (Basic) |
| BR | 8606308 | A | 871006 | 8745 | |
| US | 4742447 | A | 880503 | 8820 | |
| CN | 8608127 | A | 870729 | 8839 | |
| CA | 1266531 | A | 900306 | 9014 | |
| KR | 9205853 | B1 | 920723 | 9404 | |

Priority Data (CC No Date): US 819458 (860116)

Applications (CC,No,Date): KR 8610148 (861129); EP 87300112 (870108)

Language: English

EP and/or WO Cited Patents: 3.Jnl.Ref; A3...9024; No-SR.Pub

Designated States
(Regional): AT; BE; CH; DE; ES; FR; GB; IT; LI; NL; SE
Abstract (Basic): EP 229691
The accessing control method includes establishing a number of data structures in a dynamic manner in response to a supervisor call to map a file. The mapping process assigns a new segment of virtual memory to the mapped file and correlates in one data structure the virtual address of each page of data in the new segment to a disc file address where that page is stored.
A UNIX system call by an application program for a specific virtual page is handled by the page fault handler and not the UNIX kernel . Simple load and store type instructions are employed for the data transfer. @(20pp Dwg.No.1/7)@
Abstract (US): 8820 US 4742447
The I/O access control method establishes several data structures in a dynamic manner in response to a Supervisor call to map a file. The mapping process assigns a new segment of virtual memory to the mapped file and correlates, in one data structure, the virtual address of each page of data in the new segment to a disc file address where that page is actually stored. A UNIX kernel , since the application can supply the real address of the page on the disc file from the data structure that was created by the mapped page range Supervisor call .
Simple load and store type of instructions are employed for the data transfer, which avoids much of the overhead that normally accompanies conventional UNIX read and write system calls to the storage subsystem. @(22pp)@
Derwent Class: T01; R27;
Int Pat Class: G06F-007/00; G06F-009/00; G06F-012/08; G06F-012/10; G06F-013/00

12/7/5 (Item 5 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1995 Derwent Info Ltd. All rts. reserv.

004621865 WPI Acc No: 86-125208/19
XRPX Acc No: N86-092518
Capability based data processing security system has at least one CPU, memory and keys, each key providing authority to its holder to use specified part of systems resources
Patent Assignee: (KEYL-) KEY LOGIC INC
Author (Inventor): HARDY N
Number of Patents: 001
Patent Family:

| CC Number | Kind | Date | Week | |
|---|---|---|---|---|
| US 4584639 | A | 860422 | 8619 | (Basic) |

Priority Data (CC No Date): US 565194 (831223)
Abstract (Basic): US 4584639
The system includes factories for allowing two domains to share resources in a secure manner. Factories are special domains which, in combination with corresponding kernel functions, allow a first doman ( called a builder domain) to instal a program and other components in a factory for use by other domains, and then to seal the factory, thereby leaving the builder domain with no keys to the factory except a special type of entry key called a requestor key.
The holders of requestor keys can use the program in the factory by invoking the requestor key. This causes the factory to set up a new special domain for the requestor which allows the requestor to use the program in the factory to process data without being able to inspect

the program. Further, the factory mechanism includes means for the requestor to confirm that the factory includes no keys which could compromise the confidentiality of the requestor's data.

ADVANTAGE - Provides different memory fault resolution mechanisms ( called segment keeper domains) for different memory segments. Can be implemented in either hardware, firmware, software or their combination. @(48pp Dwg.No.2a/3b)@

Derwent Class: T01; R27;
Int Pat Class: G06F-001/00
?ds

| Set | Items | Description |
|---|---|---|
| S1 | 1011 | KERNEL |
| S2 | 822734 | SYSTEM |
| S3 | 36031 | CALL? ? OR CALLED OR CALLING |
| S4 | 24 | S1 AND S2 AND S3 |
| S5 | 47937 | INTERRUPT? |
| S6 | 2 | S4 AND S5 |
| S7 | 392494 | SWITCH? |
| S8 | 4 | S4 AND S7 |
| S9 | 44200 | FAULT? OR COMPLICATION? |
| S10 | 2 | S4 AND S9 |
| S11 | 6 | S8 OR S10 |
| S12 | 5 | S11 NOT S6 |
| S13 | 1589 | S1(2W)S3 OR S2(2W)S3 |
| S14 | 7 | S4 AND S13 |
| S15 | 3 | S14 NOT (S6 OR S11) |

?t 15/7/1-3

15/7/1 (Item 1 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1995 Derwent Info Ltd. All rts. reserv.

009643597 WPI Acc No: 93-337146/42
XRPX Acc No: N93-260537
Entity management system with remote call feature - has user-interface presentation modules, device interface access modules and function definition modules, with procedure call management
Patent Assignee: (DIGI ) DIGITAL EQUIP CORP
Author (Inventor): STRUTT C; SWIST J A
Number of Patents: 002
Number of Countries: 017
Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| WO | 9320508 | A1 | 931014 | 9342 | (Basic) |
| EP | 587880 | A1 | 940323 | 9412 | |

Priority Data (CC No Date): US 864802 (920407)
Applications (CC,No,Date): EP 93912191 (930402); WO 93US3402 (930402); WO 93US3402 (930402)
Language: English
EP and/or WO Cited Patents: 2.Jnl.Ref; EP 414624
Designated States
(National): JP
(Regional): DE; FR; GB; IT; AT; BE; CH; DK; ES; GR; IE; LU; MC; NL; PT; SE
Filing Details: EP0587880 Based on WO 9320508
Abstract (Basic): WO 9320508 A

The management system has a director kernel facilitates communication between a user of the system and managed entities through management modules located on different physical systems.

A management function of forwarding a user-initiated procedure call to the entity, and performing a desired function in relation to it, is achieved by communication between presentation modules (22), function modules (24) and access modules (26).

Two types of remote procedure calls (40,42) are included. An information manager (40) forwards procedure calls for invoking primitive functions, each on a single managed entity. A dispatcher (42) involves higher-level functions relating to user-defined domains of multiple managed entities.

USE/ADVANTAGE - In management of dispersed, complex systems. Manages entities distributed over multiple physical systems and multiple geographical locations.

Dwg.5/7

Derwent Class: T01;
Int Pat Class: G06F-009/40

15/7/2 (Item 2 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1995 Derwent Info Ltd. All rts. reserv.

008201635 WPI Acc No: 90-088636/12
XRPX Acc No: N90-068276

Supporting dynamic system calls - adding to operating system simply by loading kernel extension or program exporting entry point as SVC; SUPERVISION CALL

Patent Assignee: (ANON ) ANONYMOUS
Number of Patents: 001
Patent Family:

| CC | Number | Kind | Date | Week | |
|---|---|---|---|---|---|
| RD | 310003 | A | 900210 | 9012 | (Basic) |

Priority Data (CC No Date): RD 90310003 (900220)
Applications (CC,No,Date): RD 90------ (900220)
Abstract (Basic): RD 310003

The loader creates the appropriate system call table entry for the new system call . The importing program does not know the SVC number for a system call or even if a system call requires a change in protection domains.

The system call handler is really a set of procedures, one for each type of cross domain call supported. The loader resolves the caller to the system call through one of these handlers based on the type of call , the protection domain of the caller, and the protection domain of the system call .

ADVANTAGE - Treating system call as call allows system call handler to decrease amount of state that has to be save and restored across system calls and decrease the overhead of executing a system call . @(-pp Dwg.No.0/0)@

Derwent Class: T01; R27;
Int Pat Class: G06F-000/01

15/7/3 (Item 3 from file: 351)
DIALOG(R)File 351:DERWENT WPI
(c)1995 Derwent Info Ltd. All rts. reserv.

007948135 WPI Acc No: 89-213247/29
XRPX Acc No: N89-162477

Avoiding need to rebuild programs due to system call interception - allowing program to move from user to kernel level or vice-versa by changing trap numbers

Patent Assignee: (ANON ) ANONYMOUS
Number of Patents: 001
Patent Family:

| CC Number | Kind | Date | Week | |
|---|---|---|---|---|
| TP 68907 | A | 890625 | 8929 | (Basic) |

Priority Data (CC No Date): TP 8968907 (890620)
Abstract (Basic): TP 68907

A system call implementation scheme has been developed to allow a program to move from user to kernel level or rice versa by changing trap numbers. This ensures that the program remains intact despite system call interceptions. The scheme also avoids user/kernel trapping overhead by permitting system calls to run at user level. @(Dwg.No. 0/0)@

Derwent Class: T01; R27;
Int Pat Class: G06F-000/01
?
SYSTEM:OS - DIALOG OneSearch
File 2:INSPEC 1969-1995/Feb W4
(c) 1995 Institution of Electrical Engineers
File 6:NTIS 1964-1995/Apr B1
Comp. & distr. 1994 NTIS, US Dept of Commerce
File 8:Ei Compendex*Plus(TM) 1970-1995/Apr W3
(c) 1995 Engineering Info. Inc.
File 77:Conference Papers Index 1973-1995/Jan
(c) 1995 Cambridge Sci Abs
File 108:Aerospace Database 1962-1995/Feb
(c) 1995 AIAA
File 144:Pascal 1973-1994/Aug
(c) 1995 INIST/CNRS
File 434:SciSearch(R) 1974-1995/Feb W2
(c) 1995 Inst for Sci Info

| Set | Items | Description |
|---|---|---|

?
ds

| Set | Items | Description |
|---|---|---|
| S1 | 32578 | KERNEL |
| S2 | 4089049 | SYSTEM |
| S3 | 285334 | CALL? ? OR CALLED OR CALLING |
| S4 | 869 | S1 AND S2 AND S3 |
| S5 | 12538 | S1(2W)S3 OR S2(2W)S3 |
| S6 | 256 | S4 AND S5 |
| S7 | 46215 | INTERRUPT? OR FAULT? OT COMPLICATION? |
| S8 | 485603 | S7 OR FAULT? OR COMPLICATION? |
| S9 | 37 | S6 AND S8 |
| S10 | 21 | S6 AND SWITCH? |
| S11 | 52 | S9 OR S10 |
| S12 | 49 | S11 NOT PY=1994:1995 |
| S13 | 33 | RD S12 (unique items) |

?t 13/7/1-33

13/7/1 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

4785148 INSPEC Abstract Number: C9411-6150J-027
Title: Cohabitation and cooperation of Chorus and MacOS
Author(s): Bac, C.; Garnier, E.

Author Affiliation: Institut Nat. des Telecommun., Evry, France
Conference Title: Proceedings of the USENIX Symposium on Microkernels and Other Kernel Architectures p.61-71
Publisher: USENIX Assoc, Berkeley, CA, USA
Publication Date: 1993 Country of Publication: USA 140 pp.
Conference Title: Proceedings of the USENIX Symposium on Microkernels and Other Kernel Architectures
Conference Date: 20-21 Sept. 1993 Conference Location: San Diego, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: This paper describes experimental work on cohabitation and cooperation between a distributed operating system (Chorus) and an event driven system (MacOS). Our aims were to exploit the graphical and the musical capabilities of Macintosh hardware and software directly from Chorus applications, while minimizing our efforts in the field of device drivers and hardware interfaces. The work was carried out in four major stages. The first stage was to port the Chorus kernel on the Macintosh hardware. In the second stage we changed the way Chorus managed the hardware in order to keep the MacOS system alive. Conversely, we modified slightly the way Chorus was booted so as to present it as an application to MacOS. This led us to the third stage, which was to share system events (e.g. hardware interrupts ) between the two systems. The Chorus system allows one to have multiple functions connected to an interrupt . This feature was used to connect both an internal Chorus driver and a low level function to an interrupt . The low level function leads to the MacOS interrupt driver. The fourth stage is currently being carried out. It consists in the design and implementation of an interface permitting user level events (as system calls ) to cross the borders of the two systems. This paper describes each stage and draws lessons about system software cohabitation and reusability. (11 Refs)

13/7/2 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC


4584882 INSPEC Abstract Number: C9403-6150J-036
Title: Exploiting in- kernel data paths to improve I/O throughput and CPU availability
Author(s): Fall, K.; Pasquale, J.
Author Affiliation: California Univ., San Diego, CA, USA
Conference Title: USENIX Association. Proceedings of the Winter 1993 USENIX Conference p.327-33
Publisher: USENIX Assoc, Berkley, CA, USA
Publication Date: 1993 Country of Publication: USA x+530 pp.
Conference Sponsor: USENIX
Conference Date: 25-29 Jan. 1993 Conference Location: San Diego, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: The authors present the motivation, design, implementation, and performance evaluation of a UNIX kernel mechanism capable of establishing fast in- kernel data pathways between I/O objects. A new system call , splice() moves data asynchronously and without user-process intervention to and from I/O objects specified by file descriptors. Performance measurements indicate improved I/O throughput and increased CPU availability attributable to data copying and context switch overhead. (11 Refs)

13/7/3 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC


4433959 INSPEC Abstract Number: C9308-6150J-014
Title: Real time computer operating systems kernels: Unix compatibility and efficient real time processing
Author(s): Scharf, A.
Journal: Elektronik vol.42, no.8 p.122, 127-34
Publication Date: 20 April 1993 Country of Publication: West Germany
CODEN: EKRKAR ISSN: 0013-5658
Language: German Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: Discusses real-time embedded processor applications, especially tasks and processes, and considers implementation in connection with various operating systems such as RMX and Unix. Multi-tasking kernels and are considered, with reference to interrupt service routines (ISR), and stack memory is referred to. Context switching is illustrated and binary and counter-type semaphore flags are reported on, with special reference to resource transfer and to priority inversion. Problems of task synchronisation and of remote kernel calls are also examined. Finally, reference is made to the LynxOS multi-platform real time computer operating system , and to VMEexec, to VxWorks and to the VRTX/OS3.0 operating system , for which a run-time C-library with 69 implemented functions plus portable services is stated to be available. (0 Refs)

13/7/4 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC


04363063 INSPEC Abstract Number: C9304-6150N-050
Title: Message-based microkernel for real-time system
Author(s): Seong Rak Rim; Yoo Kun Cho
Author Affiliation: Dept. of Comput. Eng., Seoul Nat. Univ., South Korea
Conference Title: Proceedings of the Third Workshop on Future Trends of Distributed Computing Systems (Cat. No.91TH0427-5) p.174-9
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1992 Country of Publication: USA xiii+426 pp.
ISBN: 0 8186 2755 7
U.S. Copyright Clearance Center Code: 0 8186 2755 7/92$03.00
Conference Sponsor: IEEE
Conference Date: 14-16 April 1992 Conference Location: Taipei, Taiwan
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: This paper describes the design and implementation of the basic primitives and major components of the message-based microkernel for real-time systems to find out its shortcomings and ways to improve them. The real-time OS with message-based microkernel enables a user to add or change the system services easily for special purposes. But it has rather large overhead of interrupt latency and system call due to the message copy and synchronization. In order to support true real-time performance, kernel preemption and efficient message exchange mechanism is required. (13 Refs)

13/7/5 (Item 5 from file: 2)
DIALOG(R)File 2:INSPEC

04359778 INSPEC Abstract Number: C9304-6150N-039
Title: Performance comparison of message-based kernel with monolithic kernel
Author(s): Seong Rak Rim; Dong Hee Lee; Yoo Kun Cho
Journal: Journal of the Korea Information Science Society vol.19, no.6 p.640-8
Publication Date: Nov. 1992 Country of Publication: South Korea
CODEN: HJKHDC ISSN: 0258-9125
Language: Korean Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: Since kernel functions, in the case of monolithic kernels like UNIX, are implemented as a set of procedures, it is very difficult to modify these procedures or add new features to them. One way to cope with such problems is to use a message-based kernel structured as a modular set of tasks. The authors present the design and implementation of two kernel structures. They then measure the processing time of system calls and interrupt latency time. Compared with a monolithic kernel, a message-based kernel can be implemented very easily by structuring its function as a set of independent tasks, but it takes about 4 approximately 7 more to handle system calls due to message exchange and synchronization overhead. Finally, they suggest an alternative mechanism which uses shared memory to reduce this overhead and show that the performance can be enhanced dramatically. (10 Refs)

13/7/6 (Item 6 from file: 2)
DIALOG(R)File 2:INSPEC


04282298 INSPEC Abstract Number: C9212-6150J-046
Title: Parallelizing signal handling and process management in OSF/1
Author(s): Bolinger, D.; Mangalat, S.
Author Affiliation: Encore Computer Corp., Marlborough, MA, USA
Conference Title: Proceedings of the USENIX Mach Symposium p.105-22
Publisher: USENIX Assoc, Berkeley, CA, USA
Publication Date: 1991 Country of Publication: USA 262 pp.
Conference Date: 20-22 Nov. 1991 Conference Location: Monterey, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Product Review (R)
Abstract: Release 1.0 of the OSF/1 operating system, despite its high degree of parallelization, left several dozen system calls unparallelized. The most important subsystems not converted were process management and signal handling. The authors describe the project to make these subsystems multiprocessor-efficient, and to make their system calls usable within multi-threaded tasks. After presenting background on OSF/1 and on the relevant system calls, they describe the general approach and specific changes adopted for the parallelization, and for the adaptation of Unix process-oriented abstractions to the multi-threaded programming model of OSF/1. After providing rationales for the most important choices, and comparing them to a few discarded alternatives, the authors look at how some common operations are implemented in the resulting kernel, examining the resolution of races and other synchronization problems introduced by the changes. Finally, they present data on performance improvements introduced by the project, and indicate a few possibilities for useful future development. (12 Refs)

13/7/7 (Item 7 from file: 2)

DIALOG(R)File 2:INSPEC


04281403 INSPEC Abstract Number: C9212-5440-026
Title: CTRON-specification kernel implementation for a tightly coupled multiprocessor system
Author(s): Joh, M.; Igarashi, Y.; Ozeki, T.
Author Affiliation: Oki Electric Industry Co. Ltd., Chiba City, Japan
Conference Title: Proceedings. The Eighth TRON Project Symposium (Cat. No.91TH0412-7) p.118-29
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1991 Country of Publication: USA xii+249 pp.
ISBN: 0 8186 2475 2
U.S. Copyright Clearance Center Code: 0 8186 2475 2/91/$1.00
Conference Sponsor: TRON Assoc
Conference Date: 21-27 Nov. 1991 Conference Location: Tokyo, Japan
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Experimental (X)
Abstract: The authors are developing a high-performance tightly coupled multiprocessor system called OKITRON-HP, running a CTRON-specification operating system . The system is designed for application to various types of nodes in communications networks, including database nodes and transaction processing nodes, that require levels of performance not attainable in a single-processor system . Fault -tolerance functions are built in for improved reliability. They give an overview of the OKITRON-HP architecture, then discuss solutions to the problem of shared memory access (i.e. exclusion control), the most important performance factor in a multiprocessor system . Finally, they describe the kinds of algorithms implemented in the OKITRON-HP kernel based on the results of experimental simulations. (7 Refs)

13/7/8 (Item 8 from file: 2)
DIALOG(R)File 2:INSPEC


04281397 INSPEC Abstract Number: C9212-6150J-029
Title: An experimental implementation of unified real-time operating system
Author(s): Sato, K.; Tsuboto, H.; Yamamoto, O.; Saitoh, K.
Author Affiliation: LSI Lab., Mitsubishi Electric Corp., Hyogo, Japan
Conference Title: Proceedings. The Eighth TRON Project Symposium (Cat. No.91TH0412-7) p.57-68
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1991 Country of Publication: USA xii+249 pp.
ISBN: 0 8186 2475 2
U.S. Copyright Clearance Center Code: 0 8186 2475 2/91/$1.00
Conference Sponsor: TRON Assoc
Conference Date: 21-27 Nov. 1991 Conference Location: Tokyo, Japan
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Experimental (X)
Abstract: The authors are developing a unified OS which provides superior real-time response and versatile functionality simultaneously. To accomplish both features, this OS consists of two kernels. One is called nu - kernel and has large number of system calls (about 130). The other is called p- kernel and is characterized by short interrupt masking time (maximum 15 mu sec). The nu - kernel is compressed into a task in the p- kernel in order to achieve short interrupt latency. Using facilities of inter- kernel communication and synchronization. this OS makes it possible to realize a decentralized

real-time application system by distributing p-kernels to multiple microprocessors. (5 Refs)

13/7/9 (Item 9 from file: 2)
DIALOG(R)File 2:INSPEC


04221939 INSPEC Abstract Number: C9210-5440-003
Title: Implementation issues on TRANS-RTXC on the transputer
Author(s): Thielmans, H.; Verhulst, E.
Author Affiliation: K.U. Leuven Div., PMA, Heverlee, Belgium
Conference Title: Algorithms and Architectures for Real-Time Control. Proceedings of the IFAC Workshop p.123-8
Editor(s): Fleming, P.J.; Jones, D.I.
Publisher: Pergamon, Oxford, UK
Publication Date: 1992 Country of Publication: UK xii+264 pp.
ISBN: 0 08 041699 3
Conference Sponsor: IFAC; IEE; IEEE; et al
Conference Date: 11-13 Sept. 1991 Conference Location: Bangor, UK
Language: English Document Type: Conference Paper (PA)
Treatment: Applications (A); Practical (P)
Abstract: The INMOS transputer features a FIFO based scheduler in hardware, making hard real time applications difficult to program. This was solved by the implementation of a priority based preemptive scheduler. A major improvement was obtained by changing to a distributed kernel that operates transparently for the user over the underlying network. This required the implementation of a fast packet routing service and important changes to Some of the system calls . An overview is given of the layout of the kernel , the available system calls and some performance figures. (5 Refs)

13/7/10 (Item 10 from file: 2)
DIALOG(R)File 2:INSPEC


04122077 INSPEC Abstract Number: C9205-6150J-018
Title: Basic elements for a microkernel-based operating system
Author(s): Seong Rak Rim; Yoo Kun Cho
Journal: Journal of the Korea Information Science Society vol.18, no.6 p.610-18
Publication Date: Nov. 1991 Country of Publication: South Korea
CODEN: HJKHDC ISSN: 0258-9125
Language: Korean Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: In the traditional monolithic structure of operating system like UNIX, most of system services are integrated into kernel as a set of procedures. So, it is very hard to modify the procedures to add new services. Microkernel-based operating system can cope with this difficulty by structuring the system services as a set of system servers and minimizing the kernel facilities. This paper presents the design and implementation of basic kernel elements to support an operating system model which consists of a set of independent manager tasks. Kernel facilities are minimized to provide an inter-task communication (ITC) and multiplexing of system service requests. To evaluate the performance, system call overhead and interrupt latency time is measured. Finally, its ease of service addition and extensibility to real-time and distributed system is discussed. (14 Refs)

13/7/11 (Item 11 from file: 2)
DIALOG(R)File 2:INSPEC


04067248 INSPEC Abstract Number: C9202-6150N-074
Title: Dynamic resource management in distributed systems using AI-techniques
Author(s): Garner, B.; Kutti, S.
Author Affiliation: Dept. of Comput. & Math., Deakin Univ., Geelong, Vic., Australia
Conference Title: Proceedings. The Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. IEA/AIE - 89 p.1084-5 vol.2
Publisher: ACM, New York, NY, USA
Publication Date: 1989 Country of Publication: USA 2 vol. (xxxiv+1108) pp.
ISBN: 0 89791 320 5
Conference Sponsor: Univ. Tennessee; ACM; AAAI; IEEE; et al
Conference Date: 6-9 June 1989 Conference Location: Tullahoma, TN, USA
Language: English Document Type: Conference Paper (PA)
Abstract: Summary form only given. A key aspect of the kernel design based on object-oriented architecture is the resource management function demonstrated for a general purpose distributed system called the Deakin Distributed System (DDS) (Kahn, 1982). The DDS kernel is functionally divided into upper and lower kernel . The upper kernel performs resource management at system level using a dynamic knowledge-base, whereas the lower kernel is a collection of local operating systems distributed among the nodes of DDS network. The domain objects and other scheduling objects using their accumulated database apply appropriate inferencing and reasoning actions to perform dynamic resource management. The CONF object divides the nodes hierarchically into SM ( System Manager Scheduler) level, TM (Task Manager Scheduler) level and PM (Process Manager Scheduler) level objects. The significance of the achievements of this approach seem to be: novel design of the resource management kernel using an interactive knowledge-based mechanism (i.e. the SYSTEM MAP concept); automatic distribution of jobs and support for multiple goals at the same time; load balancing without process migration; automatic reconfiguration of the resource control management as the distributed system grows or shrinks physically; unlimited scaling in network size; and automatic reorganization of the control structure during system interruption (i.e. fault tolerance). (7 Refs)

13/7/12 (Item 12 from file: 2)
DIALOG(R)File 2:INSPEC


04039677 INSPEC Abstract Number: C9201-6150J-060
Title: Fault -tolerant computing based on Mach
Author(s): Babaoglu, O.
Author Affiliation: Dept. of Math., Bologna Univ., Italy
Conference Title: USENIX Workshop Proceedings. Mach p.185-99
Publisher: USENIX, Berkeley, CA, USA
Publication Date: 1990 Country of Publication: USA 199 pp.
Conference Date: 4-5 Oct. 1990 Conference Location: Burlington, VT, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)

Abstract: The author considers the problem of providing automatic and transparent fault tolerance to arbitrary user computations based on the Mach operating system . Among the several alternatives for structuring such a system , he pursues the task-pair backup paradigm in detail and outlines how it might be supported by Mach. Some of the new system calls and protocols that need to be incorporated into the Mach kernel and server tasks are sketched. (23 Refs)

13/7/13 (Item 13 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

03855577 INSPEC Abstract Number: C91030060
Title: Porting UNIX to the 386: a practical approach (designing the software specification)
Author(s): Jolitz, W.F.; Jolitz, L.G.
Journal: Dr. Dobb's Journal vol.16, no.1 p.16-18, 20, 22-4, 28, 30, 32, 34, 36, 38-40, 42, 46
Publication Date: Jan. 1991 Country of Publication: USA
CODEN: DDJSDM ISSN: 0884-5395
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: The authors discussion covers the following: the reference books; hardware and software to be used; development and definition of the 386BSD specification; 386BSD port goals; microprocessor and system specification issues; 80386 memory management; segmentation; kernal linear address space overheads; per-process data structures; virtual memory address translation mechanisms; user to kernel communication primitives; Berkley UNIX virtual memory management strategy; process context descriptions; page fault and segmentation fault mechanisms; other processor faults ; system call interface; system specific (ISA) issues; physical memory map; ISA device controllers; ISA device auto configuration; interrupt priority level management; and bootstrap operation. (3 Refs)

13/7/14 (Item 14 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

03791913 INSPEC Abstract Number: C91011089
Title: Application of real time multiprocess operating system
Author(s): Pogorelc, J.; Curkovic, M.; Premzel, B.; Strucl, J.; Fekonja, I.; Jezernik, K.; Klancar, S.; Treska, B.
Author Affiliation: Maribor Univ., Yugoslavia
Journal: Elektrotehniski Vestnik vol.57, no.4 p.237-43
Publication Date: Aug.-Oct. 1990 Country of Publication: Yugoslavia
CODEN: ELVEA2 ISSN: 0013-5852
Language: Slovenian Document Type: Journal Paper (JP)
Treatment: Applications (A); Practical (P)
Abstract: The FIOS multiprocess operating system , which has been developed to provide the flexibility, performance and UNIX compatible interface needed for efficient development and implementation of parallel real-time control code, is described. The operating system is intended for sensor based control applications such as robotics, process control and manufacturing. The features of FIOS are (among others) a support for multiple general purpose processors (based on Motorola 680*0 boards with a VME bus) and I/O devices; a high performance real-time multitasking kernel , an UNIX like environment (based on Microware OS-9/68000), which

supports most standard C system and library calls , standardized interrupt and exception handlers; and a user interface which serves to download, monitor and debug code on any processor board. As an example of an actual implementation, the authors are currently using FIOS to control a robot system . (10 Refs)

13/7/15 (Item 15 from file: 2)
DIALOG(R)File 2:INSPEC


03791305 INSPEC Abstract Number: C91010591
Title: A system architecture for fault tolerance in concurrent software
Author(s): Ancona, M.; Dodero, G.; Gianuzzi, V.; Clematis, A.; Fernandez, E.B.
Author Affiliation: Dept. of Math., Genova Univ., Italy
Journal: Computer vol.23, no.10 p.23-32
Publication Date: Oct. 1990 Country of Publication: USA
CODEN: CPTRB4 ISSN: 0018-9162
U.S. Copyright Clearance Center Code: 0018-9162/90/1000-0023$01.00
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: A system architecture called the recovery metaprogram (RMP) is proposed. It separates the application from the recovery software, giving programmers a single environment that lets them use the most appropriate fault -tolerance scheme. To simplify the presentation of the RMP approach, it is assumed that the fault model is limited to faults originating in the application software, and that the hardware and kernel layers can mask their own faults from the RMP. Also, relationships between backward and forward error recovery are not considered. Some RMP examples are given, and a particular RMP implementation is described. (8 Refs)

13/7/16 (Item 16 from file: 2)
DIALOG(R)File 2:INSPEC


03601531 INSPEC Abstract Number: C90027231
Title: CHIMERA II: a real-time multiprocessing environment for sensor-based robot control
Author(s): Stewart, D.B.; Schmitz, D.E.; Khosla, P.K.
Author Affiliation: Robotics Inst., Carnegie-Mellon Univ., Pittsburgh, PA, USA
Conference Title: Proceedings. IEEE International Symposium on Intelligent Control 1989 (Cat. No.89TH0282-4) p.265-71
Publisher: IEEE Comput. Soc. Press, Washington, DC, USA
Publication Date: 1989 Country of Publication: USA xvi+613 pp.
ISBN: 0 8186 1987 2
U.S. Copyright Clearance Center Code: TH0282-4/89/0000-0265$01.00
Conference Sponsor: IEEE
Conference Date: 25-26 Sept. 1989 Conference Location: Albany, NY, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: The CHIMERA II multiprocessing environment has been developed for use in a wide variety of sensor-based robot systems. It provides the flexibility, performance, and Unix-compatible interface needed for fast development of a real-time control code. The features of CHIMERA II include support for multiple general purpose CPUs; support for multiple special

purpose CPUs and I/O devices; a real time multitasking kernel ; user definable and dynamically selectable real-time schedulers; transparent access to a host file system ; generalized and efficient interprocess and interboard communication; remote process synchronization; standardized interrupt and exception handlers; Unix-like environment, which supports most standard C system and library calls ; support for hierarchical and horizontal control architectures, such as NASREM; and a user interface which serves to download, monitor, and debug code on any processor board and serves as a terminal interface to the executing code. (12 Refs)

13/7/17 (Item 17 from file: 2)
DIALOG(R)File 2:INSPEC


03575219 INSPEC Abstract Number: C90021359
Title: Threads and input/output in the Synthesis kernel
Author(s): Masslin, H.; Pu, C.
Author Affiliation: Dept. of Comput. Sci., Columbia Univ., New York, NY, USA
Journal: Operating Systems Review vol.23, no.5 p.191-201
Publication Date: 1989 Country of Publication: USA
CODEN: OSRED8 ISSN: 0163-5980
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: The Synthesis operating system kernel combines several techniques to provide high performance, including kernel code synthesis, fine-grain scheduling, and optimistic synchronization. Kernel code synthesis reduces the execution path for frequently used kernel calls . Optimistic synchronization increases concurrency within the kernel . Their combination results in significant performance improvement over traditional operating system implementations. Using hardware and software emulating a SUN 3/160 running SUNOS, Synthesis achieves several times to several doze times speedup for UNIX kernel calls and context switch times of 21 microseconds or faster. (7 Refs)

13/7/18 (Item 18 from file: 2)
DIALOG(R)File 2:INSPEC


03304198 INSPEC Abstract Number: C89014799
Title: SPAM: a microcode based tool for tracing operating system events
Author(s): Melvin, S.W.; Patt, Y.N.
Author Affiliation: Div. of Comput. Sci., California Univ., Berkeley, CA, USA
Journal: SIGMICRO Newsletter vol.19, no.1-2 p.58-9
Publication Date: June 1988 Country of Publication: USA
CODEN: SIGMDJ ISSN: 0163-5751
U.S. Copyright Clearance Center Code: 0163-5751/87/0012/0168$1.50
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: The authors have developed a tool called SPAM (for System Performance Analysis using Microcode), based on microcode modifications to a VAX 8600, that traces operating system events as a side-effect to normal execution. This trace of interrupts , exceptions, system calls and context switches can then be processed to analyze operating system behavior for the purpose of debugging, tuning

or development. SPAM allows measurements to be made on a fully operating UNIX system with little perturbation (typically less than 10%) and without the need for modifying the kernel . (1 Refs)

13/7/19 (Item 19 from file: 2)
DIALOG(R)File 2:INSPEC


03248315 INSPEC Abstract Number: C88064466
Title: General purpose transaction support features for the UNIX operating system
Author(s): Marcie, S.G.; Holt, R.L.
Author Affiliation: NCR Corp., E&M Columbia, W Columbia, SC, USA
Conference Title: EUUG UNIX Around the World. Proceedings of the Spring 1988 EUUG Conference p.179-84
Editor(s): Das, S.K.
Publisher: Eur. UNIX Syst. User Group, Buntingford, UK
Publication Date: 1988 Country of Publication: UK 325 pp.
ISBN: 0 9513181 0 1
Conference Sponsor: Eur. UNIX Syst. User Group
Conference Date: 11-15 April 1988 Conference Location: London, UK
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Describes the features of NCR's general purpose transaction facility (GPTF), an extension to NCR's implementation of UNIX System V for the TOWER supermicrocomputer. Timer signals with millisecond resolution are presented. Performance of process synchronization and interprocess communication is improved via a set of semaphore primitives which executes in the user program environment and operates on structures which exist in standard UNIX System V shared memory. A scheduler is described which reduces process switching latency and provides process scheduling among both realtime and timesharing priority classes. Additionally, a mechanism is provided to lock a process in memory so that it is immune to paging. Scheduling latency is reduced through voluntary preemption within the kernel . A novel disk I/O scheduler provides the ability to schedule disk requests according to process priority, seek distance, or some configurable combination of both parameters. User access to the transaction processing facilities is provided via a set of system calls and shell commands. A user friendly interface is provided to allow a superuser to control such access. (3 Refs)

13/7/20 (Item 20 from file: 2)
DIALOG(R)File 2:INSPEC


03242901 INSPEC Abstract Number: C88064403
Title: Understanding device drivers in Operating System /2
Author(s): Mizell, A.M.
Author Affiliation: Div. of Entry Syst., IBM, Boca Raton, FL, USA
Journal: IBM Systems Journal vol.27, no.2 p.170-84
Publication Date: 1988 Country of Publication: USA
CODEN: IBMSA7 ISSN: 0018-8670
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: To meet its design goals for multitasking, Operating System /2 requires a device driver architecture for interrupt -driven device management. A device driver in OS/2 is affected by the new architecture both in its structure and in its relationship to the system . An OS/2

device driver contains components, such as the strategy routine and hardware interrupt handler, which have well-defined responsibilities. The basic form of these components is a FAR CALL /FAR RETURN model. The operating system calls the device driver components to handle certain types of events, such as an application I/O request or a device interrupt . In responding to these events, an OS/2 device driver must cooperate with the operating system to preserve system responsiveness by helping to manage the multitasking of concurrent activities. Since OS/2 uses uses both the real mode and the protected mode of the system processor to support DOS and OS/2 applications, respectively, the components of an OS/2 device driver must execute in both modes. In this manner, an OS/2 device driver can be viewed as an installable extension of the Operating System /2 kernel . Comparisons between IBM Personal Computer DOS and Operating System /2 are drawn to illustrate differences between device management and device driver architecture. (6 Refs)

13/7/21 (Item 21 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

03225178 INSPEC Abstract Number: C88058504
Title: The Synthesis kernel
Author(s): Pu, C.; Massalin, H.; Ioannidis, J.
Author Affiliation: Columbia Univ., New York, NY, USA
Journal: Computing Systems vol.1, no.1 p.11-32
Publication Date: Winter 1988 Country of Publication: USA
CODEN: CMSYE2 ISSN: 0895-6340
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: The Synthesis distributed operating system combines efficient kernel calls with a high-level, orthogonal interface. The key concept is the use of a code synthesizer in the kernel to generate specialized (thus short and fast) kernel routines for specific situations. The authors have three methods of synthesizing code: factoring invariants to bypass redundant computations; collapsing layer to eliminate unnecessary procedure calls and context switches ; and executable data structures to shorten data structure traversal time. A simple model of computation called a synthetic machine supports parallel and distributed processing. The interface to synthetic machine consists of six operations on four kinds of objects. This combination of a high-level interface with the code synthesizer avoids the traditional trade-off in operating systems between powerful interfaces and efficient implementations. (16 Refs)

13/7/22 (Item 22 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

03186184 INSPEC Abstract Number: C88046650
Title: An interface providing portability for operating system kernels: the BIGSAM ideal machine
Author(s): Millard, B.R.; Miller, D.S.; Barrett, T.J.
Author Affiliation: Arizona State Univ., Tempe, AZ, USA
Conference Title: Seventh Annual International Phoenix Conference on Computers and Communications. 1988 Conference Proceedings (Cat. No.TH0188-3) p.234-9
Publisher: IEEE Comput. Soc. Press, Washington, DC, USA

Publication Date: 1988 Country of Publication: USA xxi+518 pp.
ISBN: 0 8186 0830 7
U.S. Copyright Clearance Center Code: 0896-582X/87/0000-0234$01.00
Conference Sponsor: IEEE; Arizona State Univ
Conference Date: 16-18 March 1988 Conference Location: Scottsdale, AZ, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: The design of a kernel /machine interface, which provides portability for an operating system kernel , and its implementation in the BIGSAM distributed operating system are discussed. The interface, consisting mostly of a set of procedure calls and the routines that implement them, provides an abstract or ideal machine on which the rest of the kernel runs. The basic concept of an ideal machine and its rationale as an aid to application and system portability is presented. Tradeoffs in designing an ideal machine and the principal areas of the interface are discussed. This is followed by a detailed examination of the three principal sets of ideal machine interfaces: ideal devices, ideal memory management and other ideal machine interfaces which include process/processor management and handling of system calls , interrupts and exceptions. The current status of the BIGSAM ideal machine is given. (18 Refs)

13/7/23 (Item 23 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

03177881 INSPEC Abstract Number: B88046833, C88042421
Title: I-TEST: integrated testing expert system for trunks
Author(s): Liu, D.-M.D.; Pelz, D.A.
Author Affiliation: Bell Commun. Res. Inc., Red Bank, NJ, USA
Conference Title: GLOBECOM Tokyo '87. IEEE/IECE Global Telecommunications Conference 1987. Conference Record (Cat. No.87CH2520-5) p.1825-8 vol.3
Publisher: IEEE, New York, NY, USA
Publication Date: 1987 Country of Publication: USA 3 vol. xxx+2174 pp.
U.S. Copyright Clearance Center Code: CH2520-5/87/0000-1825$01.00
Conference Sponsor: IEEE; Inst. Electron., Inf. & Commun. Eng.; Found. Adv. Int. Sci
Conference Date: 15-18 Nov. 1987 Conference Location: Tokyo, Japan
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: A prototype trunk testing expert system is described. Called the integrated testing expert system for trunks (I-TEST), it mechanizes and automates many of the conventional manual testing procedures in a switching control center (SCC) of a Bell Operating Company/Information Distribution Company (BOC/IDC). I-TEST was developed in the INTELEWINDOWS environment. This environment includes a windowing system , a graphical kernel system (GKS), and a set of high-performance portable C-programming-language-based AI tools. It combines traditional structured procedural knowledge and a typical rule-based production system into one inference mechanism to manage a complex trunk-testing environment. The modular design and generic nature of I-TEST allow it to be easily reconfigured to include test access to additional operations systems or network elements. (4 Refs)

13/7/24 (Item 24 from file: 2)
DIALOG(R)File 2:INSPEC

03146376 INSPEC Abstract Number: C88035611
Title: The DUNIX distributed operating system
Author(s): Litman, A.
Author Affiliation: Bell Commun. Res., Morristown, NJ, USA
Journal: Operating Systems Review vol.22, no.1 p.42-51
Publication Date: Jan. 1988 Country of Publication: USA
CODEN: OSRED8 ISSN: 0163-5980
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: DUNIX is an operating system that integrates several computers, connected by a packet switching network, into a single UNIX machine. As far as the users and their software can tell, the system is a single large computer running UNIX. This illusion is created by cooperation of the computers' kernels. The kernels' mode of operation is novel. The software is procedure call oriented. The code that implements a specific system call does not know whether the object in question (the file) is local or remote. That uniformity makes the kernel small and easy to maintain. The system behaves gracefully under subcomponents' failures. Users which do not have objects in a given computer are not disturbed when that computer crashes. The system administrator may switch a disk from a 'dead' computer to a healthy one, and remount the disk under the original path-name. After the switch , users may access files in that disk via the same old names. DUNIX exhibits surprisingly high performance. For a compilation benchmark, DUNIX is faster than 4.2 BSD, even if in the DUNIX case all the files in question are remote. (13 Refs)

13/7/25 (Item 25 from file: 2)
DIALOG(R)File 2:INSPEC


03088921 INSPEC Abstract Number: B88021654, C88019468
Title: The packet filter: an efficient mechanism for user-level network code
Author(s): Mogul, J.C.; Rashid, R.F.; Accetta, M.J.
Author Affiliation: Digital Equipment Corp., Western Res. Lab., Littleton, MA, USA
Journal: Operating Systems Review vol.21, no.5 p.39-51
Publication Date: 1987 Country of Publication: USA
CODEN: OSRED8 ISSN: 0163-5980
U.S. Copyright Clearance Center Code: 089791-242-X/87/0011/0039$1.50
Conference Title: Eleventh ACM Symposium on Operating Systems Principles
Conference Sponsor: ACM; Microelectron. & Comput. Technol. Corp
Conference Date: 9-11 Nov. 1987 Conference Location: Austin, TX, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: Code to implement network protocols can be either inside the kernel of an operating system or in user-level processes. Kernel -resident code is hard to develop, debug, and maintain, but user-level implementations typically incur significant overhead and perform poorly. The performance of user-level network code depends on the mechanism used to demultiplex received packets. Demultiplexing in a user-level process increases the rate of context switches and system calls , resulting in poor performance. Demultiplexing in the kernel eliminates unnecessary overhead. The paper describes the packet filter, a kernel -resident, protocol-independent packet demultiplexer. Individual

user processes have great flexibility in selecting which packets they will receive. Protocol implementations using the packet filter perform quite well, and have been in production use for several years. (23 Refs)

13/7/26 (Item 26 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

02967187 INSPEC Abstract Number: C87054773
Title: A user-level network file system in command interpreter
Author(s): Abdullah, N.; Juang, J.
Author Affiliation: Dept. of Electr. Eng. & Comput. Sci., Northwestern Univ., Evanston, IL, USA
Conference Title: IEEE Computer Society Office Automation Symposium (Cat. No.87CH2414-1) p.68-75
Publisher: IEEE Comput. Soc. Press, Washington, DC, USA
Publication Date: 1987 Country of Publication: USA xi+319 pp.
ISBN: 0 8186 0770 X
Conference Sponsor: IEEE
Conference Date: 27-29 April 1987 Conference Location: Gaithersburg, MD, USA
Language: English Document Type: Conference Paper (PA)
Treatment: General, Review (G); Practical (P)
Abstract: The authors investigate alternative approaches for implementing a network file system . Central to a network file system is a mechanism to determine whether a request refers to a local file or a remote file. By tracing the possible flows of file access as they pass through the operating system from user's end to disk storage, the authors identify three phases that are suitable for implementing the mechanism. They are the command interpreter, library functions for file system calls , and the kernel of the operating system . Implementing a network file system in a command interpreter is described in the context of the Unix operating system . It is then compared against the other two approaches in terms of their design complexity, installation efforts, capabilities, and performance. Such an implementation is the simplest one, and can offer the best performance when files are small and access frequently. It can further be improved by a file-caching scheme. In addition, the resulting network file system can be installed as a command, and allows users to switch back and forth between the usual file system and the network file system . (21 Refs)

13/7/27 (Item 27 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 1995 Institution of Electrical Engineers. All rts. reserv.

02936116 INSPEC Abstract Number: C87048727
Title: Configuration of the CTRON kernel
Author(s): Ohkubo, T.; Wasano, T.; Kogiku, I.
Author Affiliation: Telecommun. Networks Labs., NTT, Tokyo, Japan
Journal: IEEE Micro vol.7, no.2 p.33-44
Publication Date: April 1987 Country of Publication: USA
CODEN: IEMIDZ ISSN: 0272-1732
U.S. Copyright Clearance Center Code: 0272-1732/87/0400-0033$01.00
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: Networks of TRON operating systems that will incorporate industrial machines, switching nodes, communication service nodes, information processing service nodes, terminal nodes, business workstations

and intelligent objects are considered. The operating system requirements for implementing such networks are investigated, and a proposed operating system called CTRON which incorporates mechanisms to satisfy these requirements is discussed, focusing on a model for the kernel of CTRON. (10 Refs)

13/7/28 (Item 28 from file: 2)
DIALOG(R)File 2:INSPEC


02843250 INSPEC Abstract Number: B87020276, C87017660
Title: DUNIX-a distributed UNIX system
Author(s): Litman, A.
Author Affiliation: Bell Commun. Res., Morristown, NJ, USA
Conference Title: EUUG Autumn '86 Conference Proceedings p.23-31
Publisher: Eur. UNIX Syst. User Group, Buntingford, UK
Publication Date: 1986 Country of Publication: UK 499 pp.
Conference Date: 22-25 Sept. 1986 Conference Location: Manchester, UK
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: DUNIX is an operating system that integrates several computers, connected by a packet switching network, into a single UNIX machine. As far as the users and their software can tell, the system is a single large computer running UNIX. This illusion is created by cooperation of the computer's kernels. The kernels' mode of operation is novel. The code that implements a specific system call (e.g. open) does not know whether the object in question (the file) is local or remote. That uniformity makes the kernel small and easy to maintain. The system behaves gracefully under subcomponents' failures. Users which do not have objects (tty, files, processes) in a given computer are not disturbed when that computer crashes. (6 Refs)

13/7/29 (Item 29 from file: 2)
DIALOG(R)File 2:INSPEC


02427815 INSPEC Abstract Number: B85025493, C85019892
Title: Resourceful debugger sifts through faults in IEEE-802.3 LANs
Author(s): Murray, D.
Author Affiliation: Excelan Inc., San Jose, CA, USA
Journal: Electronic Design vol.33, no.2 p.173-80
Publication Date: 24 Jan. 1985 Country of Publication: USA
CODEN: ELODAW ISSN: 0013-4872
Language: English Document Type: Journal Paper (JP)
Treatment: Applications (A); Practical (P); Product Review (R)
Abstract: A system called Nutcracker is designed both to streamline the development of hardware and software products and to assist the end user in managing and maintaining the network. Packaged as an integrated workstation, the Nutcracker is built with an 8086 CPU with 960 kbytes of RAM, a CRT console, a 20-Mbyte Winchester disk, a 600-kbyte floppy-disk drive, a graphics printer, and special network hardware. A full IEEE-802.3 controller with 512 kbytes of buffer space is put to work as well as a range of instrumentation features and enhancements. The Nutcracker software consists of a multitasking kernel , a file system , an object system , a menu-driven user interface, and drivers to control the hardware. Functionally, the hardware and software break down into four logical subsystems: the acceptor, the injector, the tracer, and the statistician. Interacting, those subsystems allow the user to generate

and observe traffic on the network in a tightly controlled manner. (0 Refs)

13/7/30 (Item 1 from file: 6)
DIALOG(R)File 6:NTIS
Comp. & distr. 1994 NTIS, US Dept of Commerce. All rts. reserv.

1437184 NTIS Accession Number: AD-A215 958/0/XAB
CHIMERA II: A Real-Time UNIX-Compatible Multiprocessor Operating System for Sensor-Based Control Applications
(Technical rept)
Stewart, D. B. ; Schmitz, D. E. ; Khosla, P. K.
Carnegie-Mellon Univ., Pittsburgh, PA. Robotics Inst.
Corp. Source Codes: 005343035; 412463
Report No.: CMU-RI-TR-89-24
Sep 89 37p
Languages: English
Journal Announcement: GRAI9008
NTIS Prices: PC A03/MF A01
Country of Publication: United States
This paper describes the CHIMERA II multiprocessing operating system, which has been developed to provide the flexibility, performance, and UNIX-compatible interface needed for fast development and implementation of parallel real-time control code. The operating system is intended for sensor-based control applications such as robotics, process control, and manufacturing. The features of CHIMERA II include support for multiple general purpose CPUs; support for multiple special purpose processors and I/O devices; a high performance real-time multitasking kernel; user redefinable dynamic real-time schedulers; a UNIX-like environment, which supports most standard C system and library calls; standardized interrupt and exception handlers; and a user interface which serves to download, monitor, and debug code on any processor board, and serves as a terminal interface to the executing code. CHIMERA II also offers an attractive set of interprocessor communication features. The system-level express mail facility provides transparent access to a host file system and remote devices, and provides the basis for implementing user-level interprocessor communication. Application programmers have the choice of using shared memory, message passing, remote semaphores, or other special synchronization primitives for communicating between multiple processors. (kr)

13/7/31 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex*Plus(TM)
(c) 1995 Engineering Info. Inc. All rts. reserv.

03626598 E.I. No: EIP93030721653
Title: Multiprocessor based real-time data acquisition systems
Author: Noriega, Gerardo
Corporate Source: RMS Instruments Ltd, Mississauga, Ont, Can
Conference Title: 28th International Telemetering Conference - ITC/USA/92
Conference Location: San Diego, CA, USA
Sponsor: Int Foundation for Telemetering
E.I. Conference No.: 17861
Source: International Telemetering Conference (Proceedings) v 28 1992. Publ by Int Foundation for Telemetering, Woodland Hills, CA, USA,Ont. p 87-97
Publication Year: 1992
CODEN: ITCOD6 ISSN: 0884-5123 ISBN: 1-55617-386-5

Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications)
Journal Announcement: 9306W4
Abstract: Equipment for data collection and recording has widespread use in a variety of engineering applications. This paper deals with the use of multiprocessor-based architectures in digital data acquisition systems, emphasizing advantages in terms of flexibility and overall system throughput, and the characteristics of the embedded operating system. An overview of the basic architecture of typical data acquisition systems is first presented, followed by a description of a multiprocessing architecture for data acquisition in real-time environments where multiple sampling rates are employed to monitor analog and digital data from different sources. Software and hardware techniques are covered, including the multiplexing of analog signals, digital signal processing use of masking techniques in the processing of serial data streams, and the use of multi-point buses for communications with peripheral devices. The characteristics of a real-time multi-tasking operating system are analysed. This is the core of the software in any data acquisition system which must meet real-time constraints. In turn, the core of the operating system is the real-time kernel. Emphasis is put into the organization of the kernel, covering issues such as kernel primitives, service calls, interrupt service routines, process scheduling, memory management, and communications and synchronization between processes. (Author abstract) 10 Refs.

13/7/32 (Item 2 from file: 8)
DIALOG(R)File 8:Ei Compendex*Plus(TM)


03622345 E.I. No: EIP93020714994
Title: Message-based microkernel for real-time system
Author: Rim, Seong Rak; Cho, Yoo Kun
Corporate Source: Seoul Natl Univ, Seoul, South Korea
Conference Title: Proceedings of the Third Workshop on Future Trends of Distributed Computing Systems
Conference Location: Taipei, Taiwan
Sponsor: IEEE
E.I. Conference No.: 17752
Source: Proceedings of the Third Workshop on Future Trends of Distributed Computing Systems Proc Third Workshop Future Trends Distrib Comput Syst 1992. Publ by IEEE, Computer Society, Los Alamitos, CA, USA. p 174-179
Publication Year: 1992
ISBN: 0-8186-2755-7
Language: English
Document Type: CA; (Conference Article) Treatment: X; (Experimental)
Journal Announcement: 9306W3
Abstract: This paper describes a design and implementation of the basic primitives and major components of the message-based microkernel for real-time systems to find out it's shortcomings and ways to improve them. Through our experience, the real-time OS with message-based microkernel enables a user to add or change the system services easily for special purposes. But it has rather large overhead of interrupt latency and system call due to the message copy and synchronization. In order to support true real-time performance, kernel preemption and efficient message exchange mechanism is required. (Author abstract) 13 Refs.

13/7/33 (Item 3 from file: 8)
DIALOG(R)File 8:Ei Compendex*Plus(TM)

03045900 E.I. Monthly No: EI9104046414
Title: Aplikacija vecprocesnega operacijskega sistema za delo v realnem casu.
Title: Application of real time multiprocess operating system .
Author: Pogorelc, Janez; Curkovic, Milan; Premzel, Branko; Strucl, Joze; Fekonja, Iztok; Jezernik, Karel; Klancar, Srecko; Treska, Branko; Ljubljana
Source: Elektrotehniski Vestnik v 57 n 4 Aug-Oct 1990 p 237-243
Publication Year: 1990
CODEN: ELVEA2 ISSN: 0013-5852
Language: Slovenian
Document Type: JA; (Journal Article) Treatment: A; (Applications); G; (General Review)
Journal Announcement: 9104
Abstract: In this paper the FIOS multiprocess operating system , which has been developed to provide the flexibility, performance, and UNIX compatible interface needed for efficient development and implementation of parallel real-time control code, is decribed. The operating system is intended for sensor based control applications such as robotics, process control and manufacturing. The features of FIOS are (among others) a support for multiple general purpose processors (based on Motorola 680 multiplied by 0 boards with a VME bus) and I/O devices; a high performance real-time multitasking kernel , an UNIX like environment (based on Microware OS-9/68000), which supports most standard C system and library calls , standardized interrupt and exception handlers; and a user interface which serves to down-load, monitor and debug code on any processor board. As an example of an actual implementation, we are currently using FIOS to control a robot system . (Author abstract) 10 Refs.

?s intercept? or exception?
22998 INTERCEPT?
65486 EXCEPTION?
S14 88412 INTERCEPT? OR EXCEPTION?
?ds

| Set | Items | Description |
|---|---|---|
| S1 | 32578 | KERNEL |
| S2 | 4089049 | SYSTEM |
| S3 | 285334 | CALL? ? OR CALLED OR CALLING |
| S4 | 869 | S1 AND S2 AND S3 |
| S5 | 12538 | S1(2W)S3 OR S2(2W)S3 |
| S6 | 256 | S4 AND S5 |
| S7 | 46215 | INTERRUPT? OR FAULT? OT COMPLICATION? |
| S8 | 485603 | S7 OR FAULT? OR COMPLICATION? |
| S9 | 37 | S6 AND S8 |
| S10 | 21 | S6 AND SWITCH? |
| S11 | 52 | S9 OR S10 |
| S12 | 49 | S11 NOT PY=1994:1995 |
| S13 | 33 | RD S12 (unique items) |
| S14 | 88412 | INTERCEPT? OR EXCEPTION? |
| S15 | 11 | S6 AND S14 |
| S16 | 6 | RD S15 (unique items) |
| S17 | 0 | S16 NOT S13 |

?
SYSTEM:OS - DIALOG OneSearch
File 35:Dissertation Abstracts Online 1861-1995/Jan
(c) 1994 UMI
File 202:Information Science Abs. 1966-1994/Oct
(c) 1995 IFI/Plenum Data Corp.

```
      Set  Items  Description
      ---  -----  -----------
?
ds

Set     Items   Description
S1       2087   KERNEL
S2     172402   SYSTEM
S3      37621   CALL? ? OR CALLED OR CALLING
S4        102   S1 AND S2 AND S3
S5       1231   S1(2W)S3 OR S2(2W)S3
S6         30   S4 AND S5
S7       2890   INTERRUPT? OR FAULT? OT COMPLICATION?
S8      11533   S7 OR FAULT? OR COMPLICATION?
S9          2   S6 AND S8
S10         3   S6 AND SWITCH?
S11         4   S9 OR S10
S12         4   S11 NOT PY=1994:1995
S13         4   RD S12 (unique items)
S14     11547   INTERCEPT? OR EXCEPTION?
S15         2   S6 AND S14
S16         2   RD S15 (unique items)
S17         1   S16 NOT S13
?s s13 or s17
                4  S13
                1  S17
     S18        5  S13 OR S17
?t 18/7/1-5

 18/7/1      (Item 1 from file: 35)
DIALOG(R)File  35:Dissertation Abstracts Online
(c) 1994 UMI. All rts. reserv.

01244547  ORDER NO: AAD92-32050
SYNTHESIS: AN EFFICIENT IMPLEMENTATION OF FUNDAMENTAL OPERATING *SYSTEM*
SERVICES (CODE GENERATION)
  Author:  MASSALIN, HENRY
  Degree:  PH.D.
  Year:    1992
  Corporate Source/Institution:  COLUMBIA UNIVERSITY (0054)
  Adviser: CALTON PU
  Source:  VOLUME 53/06-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
           PAGE 2989.  148 PAGES

     This dissertation shows that operating systems can provide fundamental
services an order of magnitude more efficiently than traditional
implementations. It describes the implementation of a new operating
*system* *kernel*, Synthesis, that achieves this level of performance.
     The Synthesis *kernel* combines several new techniques to provide high
performance without sacrificing the expressive power or security of the
*system*. The new ideas include: (1) Run-time code synthesis--a systematic
way of creating executable machine code at runtime to optimize
frequently-used *kernel* routines--queues, buffers, context *switchers*,
*interrupt* handlers, and *system* *call* dispatchers--for specific
situations, greatly reducing their execution time. (2) Fine-grain
scheduling--a new process-scheduling technique based on the idea of
feedback that performs frequent scheduling actions and policy adjustments
(at sub-millisecond intervals) resulting in an adaptive, self-tuning
*system* that can support real-time data streams. (3) Lock-free optimistic
```

synchronization is shown to be a practical, efficient alternative to lock-based synchronization methods for the implementation of multiprocessor operating *system* kernels. (4) An extensible *kernel* design that provides for simple expansion to support new *kernel* services and hardware devices while allowing a tight coupling between the *kernel* and the applications, blurring the distinction between user and *kernel* services.

The result is a significant performance improvement over traditional operating *system* implementations in addition to providing new services.

18/7/2 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abstracts Online


01213509 ORDER NO: AAD92-11954
DESIGN, IMPLEMENTATION, AND EVALUATION OF A REAL-TIME *KERNEL* FOR DISTRIBUTED ROBOTICS (ROBOTICS, TIMIXV2)
Author: KING, ROBERT BRUCE, II
Degree: PH.D.
Year: 1991
Corporate Source/Institution: UNIVERSITY OF PENNSYLVANIA (0175)
Supervisor: INSUP LEE
Source: VOLUME 52/11-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 5930. 141 PAGES

Modern robotics applications are becoming more complex due to greater numbers of sensors and actuators. The control of such systems may require multiple processors to meet the computational demands and to support the physical distribution of the sensors and actuators. A distributed real-time *system* is needed to perform the required communication and processing while meeting application-specified timing constraints. Our research is the design and evaluation of a real-time *kernel*, *called* TimixV2, for distributed robotics applications.

TimixV2 provides threads with dynamic timing constraints, execution environments as basic units for resource allocation and memory management context, and events to signal message arrival, device *interrupts*, alarms, and exceptions. The salient features of TimixV2 are support for uniform scheduling and timely communication. TimixV2 uses the notion of consistent scheduling to uniformly schedule both application and *kernel* threads to guarantee that the application's real-time constraints are met. All device *interrupt* handlers, except the periodic clock *interrupt*, are converted to threads that are scheduled like any other thread. TimixV2's port-based message passing primitives support real-time communication by allowing individual message priorities to be used to order messages on a queue and by propagating scheduling information from a message to the associated thread on message arrival.

The *kernel* has been implemented on a distributed test-bed and evaluated with respect to distributed real-time robotics applications.

18/7/3 (Item 3 from file: 35)
DIALOG(R)File 35:Dissertation Abstracts Online


846178 ORDER NO: NOT AVAILABLE FROM UNIVERSITY MICROFILMS INT'L.
COMPUTER *SYSTEM* MEASUREMENT
Author: CARRINGTON, DAVID ATHOL
Degree: PH.D.
Year: 1984
Corporate Source/Institution: UNIVERSITY OF NEW SOUTH WALES (AUSTRALIA)

(0423)
Source: VOLUME 45/03-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 928.

This thesis consists of a study of computer *system* measurement centered on the hardware-software interface and a description of an experimental hybrid monitor *system*, Mu, designed and constructed by the author.

The role of measurement in computer performance evaluation is discussed and fundamental monitoring concepts and strategies are examined. A substantial review of previous measurement research is provided focussing on the design and application of hardware and software monitors. This chronologically organised review shows the development of this field of computer performance evaluation.

A notation is developed for specifying monitoring experiments in terms of event patterns and the corresponding measurement actions. This is an important tool since it hides many of the implementation details from the user and allows the monitor to be manipulated at a more abstract "level". The hardware design is based on the functions of matching event patterns and processing the associated event actions. A parallel implementation is necessary to keep pace with the subject *system*.

A collection of experiments demonstrating the practical application of the monitor is reported. The experiments concentrate on the performance of sections of the UNIX* operating *system* *kernel*, looking at procedure *calling* overheads, the distribution of *system* *call* activity, and context *switching*.

A discussion of computer *system* features that facilitate or impede the use of monitoring techniques offers guidance for *system* design. The point of view that consideration of measurement requirements is an important facet of design is emphasised.

18/7/4 (Item 4 from file: 35)
DIALOG(R)File 35:Dissertation Abstracts Online


789162 ORDER NO: AAD82-21630
DEBUGGING TECHNIQUES FOR COMMUNICATING, LOOSELY-COUPLED PROCESSES
Author: SMITH, EDWARD TUCKER
Degree: PH.D.
Year: 1982
Corporate Source/Institution: THE UNIVERSITY OF ROCHESTER (0188)
Source: VOLUME 43/04-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 1176. 142 PAGES

This thesis describes work done on debugging techniques and tools for communicating, loosely-coupled processes. Our work is intended to reduce the apparent complexity of large systems of communicating programs by regarding only the interprocess activities of such programs. The use of multiple, communicating processes as a model of computation allows for a very clean "cut" of what information is interesting for debugging and what is not. Our approach to debugging is to provide the user with information about how sets of these processes behave rather than what each program associated with each process does.

Our tools provide various primitives for manipulating the interprocess activities of processes. We provide nothing to access the source code of any program. Our tools include a debugger program, a mechanism to fire and execute interprocess debugging demons and the ability to obtain transcripts of interprocess activities. The debugger provides

commands for the user at a terminal for creating and manipulating individual interprocess events. Demons are an event-driven mechanism used to automatically monitor and modify interprocess events. Transcripts provide a record of interprocess events that can be replayed later.

Our debugging techniques make use of these tools to provide individual process control, communication monitoring and process testing. Process control includes the ability to create, suspend and destroy processes as well as the ability to obtain various process-related information. The communication monitoring facility monitors message traffic and can dynamically alter the contents of these messages. Process testing allows a user to isolate a process (or simulate a process) by creating and *intercepting* all message traffic in and out of a process.

This thesis also describes a debugging *system* *called* SPIDER that was built to demonstrate the above debugging tools and techniques. A multi-process *kernel* is included in SPIDER that supports communicating, loosely-coupled processes. SPIDER also includes an implementation of each of our proposed tools: a debugger program, a mechanism for firing demons and a transcriber. Several examples using SPIDER are given to show how our debugging techniques can be achieved with our debugging tools.

18/7/5 (Item 1 from file: 202)
DIALOG(R)File 202:Information Science Abs.


00144769 9004769
ISA Document Number in Printed Publication: 9005018
The Synthesis *kernel*.
Document Type: Journal Article
Author (Affiliation): Pu, C. (Columbia Univ., New York, NY); Ioannidis, J. ; Massalin, H.
Country of Affiliation: United States
Journal: Computing Systems
Publication Language(s): English
Source: Vol. 1 Issue 1 p. 11-32 Win 1988 16

This paper describes the Synthesis distributed operating *system* that combines efficient *kernel* *calls* with a high-level, orthogonal interface. The key concept is the use of a code synthesizer in the *kernel* to generate specialized routines. Three methods of synthesizing code are studied: factoring invariants to bypass redundant computations, collapsing layers to eliminate unnecessary procedure *calls* and context *switches*, and executable data structures to shorten data structure traversal time. The author also notes that the combination of high-level interface with the code synthesizer avoids the traditional trade-off in operating systems between powerful interfaces and efficient implementations.

?
SYSTEM:OS - DIALOG OneSearch
File 233:Microcomputer Abstracts(TM) 1981-1995/Feb
(c) 1995 Learned Inform.Inc.
File 237:Buyer's Guide to Micro Software(SOFT) 1993/Sep
(c) 1993 ONLINE Inc.
File 256:SoftBase:Reviews,Companies & Prods. 1995/Jan
(c) 1995 Info.Sources Inc
File 278:Microcomput.Software Guide 1995/Mar
(c) 1995 Reed Reference Publishing
File 751:Datapro Software Directory 1995/Jan
(c) 1995 McGraw-Hill, Inc.

```
      Set  Items  Description
      ---  -----  -----------
?
ds

Set      Items   Description
S1         732   KERNEL
S2       96217   SYSTEM
S3       18649   CALL? ? OR CALLED OR CALLING
S4          62   S1 AND S2 AND S3
S5         671   S1(2W)S3 OR S2(2W)S3
S6          19   S4 AND S5
S7        1201   INTERRUPT? OR FAULT? OT COMPLICATION?
S8        3046   S7 OR FAULT? OR COMPLICATION?
S9           3   S6 AND S8
S10          1   S6 AND SWITCH?
S11          3   S9 OR S10
S12          2   S11 NOT PY=1994:1995
S13          1   RD S12 (unique items)
S14       3031   INTERCEPT? OR EXCEPTION?
S15          0   S6 AND S14
S16          0   RD S15 (unique items)
S17          0   S16 NOT S13
?t 11/7/1-3

 11/7/1      (Item 1 from file: 233)
DIALOG(R)File 233:Microcomputer Abstracts(TM)
(c) 1995 Learned Inform.Inc. All rts. reserv.

0251255   91DD10-003
   Porting UNIX to the 386: The basic kernel -- Multiprogramming and
multitasking, part II
   Jolitz, William Frederick; Jolitz, Lynne Greer
   Dr. Dobb's Journal , October 1, 1991 , v16 n10 p62-72, 118+, 9 Page(s)
ISSN: 1044-789X
   The second part of a series of articles presents five C programs that
supplement a discussion on the 386BSD switching mechanisms for porting UNIX
applications to 386 machines. The first listing shows a code fragment
executed after a system call or interrupt; the second, tsleep(), is a
blocking call that sets a process sleeping and runs the other process until
the event occurs; wakeup() removes the block that stops processes sleeping
for it; listing four illustrates storing and loading of the processor
state; and listing five places the setrq() routine at the trail of the run
queue associated with the process' priority. Includes five program
listings. (tbc)

 11/7/2      (Item 1 from file: 256)
DIALOG(R)File 256:SoftBase:Reviews,Companies & Prods.
(c) 1995 Info.Sources Inc. All rts. reserv.

01003817          DOCUMENT TYPE:  Product

PRODUCT NAME:  C Executive 2.4A (003817)

JMI Software Consultants Inc (128724)
904 Sheble Ln  PO Box 481
Spring House, PA  19477  United States
TELEPHONE:  (215) 628-0840
```

RECORD TYPE: Directory

CONTACT: Edward Rathje, Pres

C Executive 2.4A is a real-time operating *system* specifically designed for time-critical applications written in C language. It provides a combination of *kernel* level services (preemptive scheduling, clock services, full I/O subsystem, resource coordination, etc.) and C language interfaces. Standard features include semaphores, events and data stream queues (not limited by 'mailbox' size). The *interrupt* driven, multi-terminal device drivers include XON/XOFF capability. C language *system* *calls* are built-in. Real-time applications requiring UNIX may be able to use C Executive instead. Embedded applications include FAA radar, oceanographic and laboratory data acquisition, cardiac monitors, PBX, process control and communications.

REVISION DATE: 920423

11/7/3 (Item 2 from file: 256)
DIALOG(R)File 256:SoftBase:Reviews,Companies & Prods.


00060428 DOCUMENT TYPE: Review

PRODUCT NAMES: BSD Unix 4.4 (902108)

TITLE: What's New in 4.4BSD
AUTHOR: McKusick, Marshall Kirk
SOURCE: UNIX Review, v12 n1 p51(4) Jan 1994 0742-3136

RECORD TYPE: Review
REVIEW TYPE: Product Analysis
GRADE: Product Analysis, No Rating

4.4BSD Unix will be the last release from the Computer Systems Research Group, due to decreased funding, and declining support from University of California. Commercially, it will be available from Berkeley Software Design, and other variants may continue to be available as freeware. This latest release includes several enhancements, including a redesigned virtual memory *system*. There are some structural *kernel* changes as well, and the *kernel* uses a new internal *system*-*call* convention, and *interrupted* *system* *calls* will no longer abort using non-local goto's. The new virtual memory implementation comes from the Mach *system*. The 4.4BSD implementation also contains a virtual file *system* interface to support multiple file systems. New tools and utilities are also offered for greater ease of use and functionality, and the Kerberos authentication software has been integrated into the software.

REVISION DATE: 940525
?
*File 275:Computer Database(TM) 1983-1995/Feb 14

*File 275: File 675 has been consolidated into File 275.

| Set | Items | Description |
|---|---|---|

?
ds

| Set | Items | Description |
|---|---|---|
| S1 | 5446 | KERNEL |
| S2 | 318845 | SYSTEM |
| S3 | 150885 | CALL? ? OR CALLED OR CALLING |
| S4 | 2861 | S1 AND S2 AND S3 |
| S5 | 7463 | S1(2W)S3 OR S2(2W)S3 |
| S6 | 673 | S4 AND S5 |
| S7 | 11179 | INTERRUPT? OR FAULT? OT COMPLICATION? |
| S8 | 28141 | S7 OR FAULT? OR COMPLICATION? |
| S9 | 338 | S6 AND S8 |
| S10 | 275 | S6 AND SWITCH? |
| S11 | 423 | S9 OR S10 |
| S12 | 381 | S11 NOT PY=1994:1995 |
| S13 | 379 | RD S12 (unique items) |
| S14 | 19465 | INTERCEPT? OR EXCEPTION? |
| S15 | 186 | S6 AND S14 |
| S16 | 185 | RD S15 (unique items) |
| S17 | 52 | S16 NOT S13 |
| S18 | 2031 | S5/AB |
| S19 | 41 | (S11 OR S15) AND S18 |
| S20 | 7729 | S8/AB |
| S21 | 10 | S19 AND S20 |

?t 21/5/1-10

21/5/1

DIALOG(R)File 275:Computer Database(TM)


01719753 SUPPLIER NUMBER: 16245543 (USE FORMAT 7 FOR FULL TEXT)
Writing serial drivers for UNIX. (includes related article on ring buffers) (Tutorial)
Wells, Bill
Dr. Dobb's Journal, v19, n15, p68(6)
Dec, 1994
DOCUMENT TYPE: Tutorial ISSN: 1044-789X LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2859 LINE COUNT: 00220

ABSTRACT: Good serial driver design for UNIX systems involves a solid understanding of control flow, the kernel interface and the serial device. A well-designed driver will have distinctive sections, including debugging and statistics functions, hardware management capabilities, state changes, a system - call interface and the ability to make declarations. The declarations section provides specific type and variable declarations, as well as a LINE_STATE enumeration specifying the overall line conditions. A circular buffer can be used to record function entry and function exit calls and a status print routine will enable the programmer to interrupt processing at any time for a hard copy of line variables.
SPECIAL FEATURES: illustration; program
FILE SEGMENT: CD File 275
DESCRIPTORS: Serial Interface; Device Driver; UNIX; Program Development Techniques; Tutorial
SIC CODES: 7371 Computer programming services

21/5/2

DIALOG(R)File 275:Computer Database(TM)

01654523 SUPPLIER NUMBER: 16294160
QNX forges ahead. (QNX Software's QNX real-time operating system ) (Product Announcement)
Varhol, Peter D.
Byte, v19, n10, p199(3)
Oct, 1994
DOCUMENT TYPE: Product Announcement ISSN: 0360-5280 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: QNX Software's QNX real-time operating system provides 32-bit performance, complies with Posix and provides the look and feel of Unix at a much lower cost. The operating system employs a pure microkernel architecture that passes messages in a way that is transparent to the network and provides impressive modularity. Kernel calls execute very quickly because the 8Kb microkernel fits into the on-chip caches of Pentium or 486 microprocessors. A compact API handles the microkernel's four main functions: interprocess communication, network communication, interrupt dispatch and process scheduling. The message-passing operating system employs blocking versions of Send, Receive and Reply function calls . A lack of commercial applications software will limit the widespread acceptance of QNX, but it will prove highly useful for real-time projects and custom software development.

SPECIAL FEATURES: illustration; chart
FILE SEGMENT: CD File 275
COMPANY NAMES: QNX Software--Product introduction
DESCRIPTORS: Operating System ; Product Introduction; Real-Time System
SIC CODES: 7372 Prepackaged software
TRADE NAMES: QNX (Operating system )-Product introduction

21/5/3
DIALOG(R)File 275:Computer Database(TM)


01488095 SUPPLIER NUMBER: 12676061
A distributed real-time operating system . (includes related article on the Hexagonal Architecture for Real-Time Systems environment) (Technical)
Shin, Kang G.; Kandlur, Dilip D.; Kiskis, Daniel L.; Dodd, Paul S.; Rosenberg, Harold A.; Indiresan, Atri
IEEE Software, v9, n5, p58(11)
Sept, 1992
DOCUMENT TYPE: Technical ISSN: 0740-7459 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

ABSTRACT: Researchers from the University of Michigan are developing a 19-node hexagonal mesh operating system , called Hexagonal Architecture for Real-Time Systems (HARTS). Two versions of HARTS are presented. One version features enhanced pSOS services to provide a distributed naming service and interprocessor communication; the other includes real-time fault -tolerant communication. HARTS' communication services are provided via protocols running on the X- kernel , which includes several levels consisting of the link-level protocol, the normal link-level protocol, the clock-synchronization protocol and the user datagram protocol, respectively. The user datagram protocol, as well as all higher level protocol modules provide services that are accessible by the user: reliable broadcasting, remote procedure calls and a real-time-channel service. Evaluation tools are also discussed.

SPECIAL FEATURES: illustration; chart; table
FILE SEGMENT: AI File 88
DESCRIPTORS: Distributed Systems; Computer Science; System Design; Real-Time System ; Operating System ; Kernel

21/5/4
DIALOG(R)File 275:Computer Database(TM)


01366811 SUPPLIER NUMBER: 08699702 (USE FORMAT 7 FOR FULL TEXT)
Supporting protected-mode applications in a DOS-based environment. (tutorial)
Duncan, Ray
Microsoft Systems Journal, v5, n4, p92(5)
July, 1990
DOCUMENT TYPE: tutorial ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2595 LINE COUNT: 00201

ABSTRACT: The Virtual Control Program Interface (VCPI), the industry standard as of Apr 1989, is the predecessor of the DOS Protected-Mode Interface (DPMI). Since it was inadequate for multitasking DOS extender applications, DPMI was developed, with a server, which unlike that of VCPI, runs at a higher privilege level than its clients. DPMI thus provides the safe execution of DOS extender protected mode applications within DOS-based multitasking environments. It addresses problems which arise when two or more high performance protected mode applications are vying for system resources. Function calls supported by DPMI fall into 7 categories: LDT management services; DOS memory management services; extended memory management services; page management services; interrupt management services; translations services; and miscellaneous services. The prototype of DPMI was developed for Microsoft Windows 3.0.
CAPTIONS: Function numbers and names of DPMI services exported for use. (table); Relationship between DOS extender, DPMI server and MS-DOS. (chart) ; DPMI return values for INT 2FH. (table)

SPECIAL FEATURES: illustration; table; chart
FILE SEGMENT: CD File 275
DESCRIPTORS: DOS Extenders; Multitasking; Extended Memory; Debugging Tools; Client/Server Architecture; User Interface; Tutorial; MS-DOS
SIC CODES: 7372 Prepackaged software
TICKER SYMBOLS: MSFT
TRADE NAMES: Microsoft Windows 3.0 (GUI)-Usage
OPERATING PLATFORM: MS-DOS

21/5/5
DIALOG(R)File 275:Computer Database(TM)


01363935 SUPPLIER NUMBER: 08615398 (USE FORMAT 7 FOR FULL TEXT)
Real-time UNIX & UNIX look-alikes. (EDN Special Report)
Small, Charles H.
EDN, v35, n12, p88(14)
June 7, 1990
ISSN: 0012-7515 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6345 LINE COUNT: 00493

ABSTRACT: The memory-management units on new 32-bit microprocessors and

their large on-board memories allow Unix to fit comfortably on single-board computers, but many Unix characteristics must be changed and others added for the operating system to be used in embedded applications. Real-time Unix systems are offered by many vendors to address the needs of embedded designers, including deterministic response to interrupts, a prioritized multitasking scheduler that can be prompted, computationally intensive tasks, fast, race-free intertask communications and control mechanisms; compact reentrant code modules, library routines and operating-system calls; a secure and fast file system, and the ability to run from ROM without terminals or offline storage and to recovery from outages quickly and safely. Several real-time Unix versions are described, including Emerge Systems' RTUX, Modcomp's REAL/IX and Lynx Real-Time Systems' Lynxos.
CAPTIONS: Major actors in a Unix system and how they interact. (chart); Selecting between dedicated, real-time executive and real-time Unix. (graph); Conventional diagrams of Unix from the software-developer perspective. (chart)

SPECIAL FEATURES: illustration; chart; graph
FILE SEGMENT: TI File 148
DESCRIPTORS: UNIX; Embedded Systems; Real-Time System; Software Design; System Design; User Need; Comparison
SIC CODES: 7372 Prepackaged software
OPERATING PLATFORM: Unix

21/5/6
DIALOG(R)File 275:Computer Database(TM)


01350440 SUPPLIER NUMBER: 08155226 (USE FORMAT 7 FOR FULL TEXT)
Is it time for VS II? It's time to stop whining and take a good look at AOS/VS II.
Horvitz, Phil
DG Review, v10, n8, p8(4)
Feb, 1990
ISSN: 1050-9127 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2055 LINE COUNT: 00144

ABSTRACT: Installation of the AOS/VS II operating system on a Data General MV computer system will probably slow performance, depending on the number of system calls made by a given application. How much slower the system will run depends on what applications are run and the kinds of system calls they execute. Programs performing many file system calls are going to suffer deterioration of performance in a fault -tolerant filing system. Applications can be tuned to run more rapidly under VS II by minimizing file system calls or by activating LDU data Caching, a new VS II feature. Users can also enhance performance by replacing lower disk drives with faster-operating ones. Networking performance is improved under AOS/VS II.
FILE SEGMENT: CD File 275
DESCRIPTORS: Operating System; Performance Improvement; Cache Memory
SIC CODES: 7372 Prepackaged software
TRADE NAMES: Data General MV-2000 (Minicomputer)-Computer programs; AOS/VS II (Operating system)-Usage

21/5/7
DIALOG(R)File 275:Computer Database(TM)

01284524 SUPPLIER NUMBER: 07190599 (USE FORMAT 7 FOR FULL TEXT)
Taking a realistic look at DOS 4.0. (Power Programming) (technical)
Duncan, Ray
PC Magazine, v8, n1, p329(4)
Jan 17, 1989
DOCUMENT TYPE: technical ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1663 LINE COUNT: 00127

ABSTRACT: The DOS modifications contained in DOS release 4.0 that impact software developer are discussed. The new system 's least significant feature is its much-discussed support for LIM EMS. In actuality DOS 4.0's EMS support is little more than a disk caching scheme that can use expanded memory. One completely new system call contained in DOS 4.0 is the interrupt 21h function 6Ch, Extended Open File. This function allows the programmer separately to specify how the function behaves if the file does not already exist. It also returns a code indicating the actual action taken by the function. DOS will not be providing any further radical changes or improvements; users looking for multitasking, virtual memory, interprocess communications and hardware-dependent graphics should look to a protected-mode environment, where the hardware forces applications programmers to follow the rules.
CAPTIONS: New DOS 4.0 DEBUG commands. (table); Interrupts 25h and 26h under DOS 4.0. (chart); DOS 4.0 boot sector layout. (table)

SPECIAL FEATURES: illustration; table; chart
FILE SEGMENT: CD File 275
COMPANY NAMES: International Business Machines Corp.--Products
DESCRIPTORS: Operating System ; MS-DOS; Software Design; Technology
SIC CODES: 7372 Prepackaged software
TICKER SYMBOLS: IBM
TRADE NAMES: MS-DOS 4.0 (Operating system )-Design and construction
OPERATING PLATFORM: MSDOS

21/5/8
DIALOG(R)File 275:Computer Database(TM)


01230501 SUPPLIER NUMBER: 06274218
Writing UNIX device drivers. (Continuation of series from two previous issues.)
Pajari, George E.
UNIX World, v5, n3, p89(5)
March, 1988
ISSN: 0739-5922 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: To install a driver, the driver code must be compiled and linked with the UNIX system kernel . Major device numbers must be assigned to each driver; the driver must be tested and debugged; partially modified variables must be protected against existing interrupt routines. One solution is that the kernel call the interrupt routine every 1-50th of a second. Another is to have the driver set a timeout when an interrupt is expected. Writing a driver is not as easy as writing an application program.
CAPTIONS: Tables for installation. (table)

SPECIAL FEATURES: illustration; table
FILE SEGMENT: CD File 275

DESCRIPTORS: UNIX; Device Driver; Interrupts

21/5/9
DIALOG(R)File 275:Computer Database(TM)


01221021 SUPPLIER NUMBER: 07209493
Understanding device drivers in Operating System -2. (technical)
Mizell, A.M.
IBM Systems Journal, v27, n2, p170(15)
June, 1988
DOCUMENT TYPE: technical ISSN: 0018-8670 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

ABSTRACT: To meet its design goals for multitasking, Operating System -2 requires a device driver architecture for interrupt -driven device management. A device driver in OS-2 is affected by the new architecture both in its structure and in its relationship to the system . An OS-2 device driver contains components, such as the Strategy Routine and Hardware interrupt Handler, which have well-defined responsibilities. The basic form of these components is a FAR CALL -FAR RETURN model. The operating system calls the device driver components to handle certain types of events, such as an application I-O request or a device interrupt . In responding to these events, an OS-2 device driver must cooperate with the operating system to preserve system responsiveness by helping to mange the multitasking of concurrent activities. Since OS-2 uses both the real mode and the protected mode of the system processor to support DOS and OS-2 applications, respectively, the components of an OS-2 device driver must execute in both modes. In this manner, an OS-2 device driver can be viewed as an installable extension for the Operating System -2 kernel . Comparisons between IBM Personal Computer DOS and Operating System -2 are drawn to illustrate differences between device management and device driver architecture.
CAPTIONS: Polled I-O. (chart); Interrupt -driven I-O. (chart); Relationship of the OS-2 device driver to the system . (chart)

SPECIAL FEATURES: illustration; chart
FILE SEGMENT: AI File 88
COMPANY NAMES: International Business Machines Corp.--Products
DESCRIPTORS: Operating System ; Device Driver; OS/2
TICKER SYMBOLS: IBM

21/5/10
DIALOG(R)File 275:Computer Database(TM)


01074169 SUPPLIER NUMBER: 00587158
Stratus Offers Unix Implementation for Mainframes.
Computerworld, v18, n46, p86
Nov. 12, 1984
DOCUMENT TYPE: product announcement ISSN: 0010-4841 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: Stratus Computer has introduced a version of the Unix system for its Continuous Processing series of fault -tolerant mainframes. The implementation, called USF, is integrated with the VOS operating system at the kernel level. It contains standard Unix System V

features such as C language, shell, commands and applications, system calls and subroutines, productivity tools, and program utilities. USF also includes VOS facilities such as demand-paged virtual memory, indexed files, record locking, access-control lists, and transparent networking. USF will be available in March 1985; a sixteen-user version costs $8,000 and the sixty-five user version costs $32,000.

FILE SEGMENT: CD File 275
DESCRIPTORS: UNIX; Operating System ; New Product; Mainframe Computer; UNIX-Like Operating Systems
TRADE NAMES: USF
OPERATING PLATFORM: UNIX; USF
?
SYSTEM:OS - DIALOG OneSearch
File 15:ABI/INFORM(R) 1971-1995/Feb W3
(c) 1995 UMI
File 16:PTS Promt(TM) 1972-1995/Mar 03
(c) 1995 Information Access Co.

| Set | Items | Description |
|---|---|---|

?
ds

| Set | Items | Description |
|---|---|---|
| S1 | 4623 | KERNEL |
| S2 | 854530 | SYSTEM |
| S3 | 719892 | CALL? ? OR CALLED OR CALLING |
| S4 | 1197 | S1 AND S2 AND S3 |
| S5 | 11856 | S1(2W)S3 OR S2(2W)S3 |
| S6 | 217 | S4 AND S5 |
| S7 | 19587 | INTERRUPT? OR FAULT? OT COMPLICATION? |
| S8 | 62667 | S7 OR FAULT? OR COMPLICATION? |
| S9 | 59 | S6 AND S8 |
| S10 | 58 | S6 AND SWITCH? |
| S11 | 84 | S9 OR S10 |
| S12 | 66 | S11 NOT PY=1994:1995 |
| S13 | 64 | RD S12 (unique items) |
| S14 | 80247 | INTERCEPT? OR EXCEPTION? |
| S15 | 31 | S6 AND S14 |
| S16 | 31 | RD S15 (unique items) |
| S17 | 17 | S16 NOT S13 |
| S18 | 8207 | S5/AB |
| S19 | 27 | (S11 OR S15) AND S18 |
| S20 | 36842 | S8/AB |
| S21 | 16 | S19 AND S20 |

? 16 S21
752445 PY=1994 : PY=1995
S22 11 S21 NOT PY=1994:1995
?t 22/7/1-11

22/7/1 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/INFORM(R)
(c) 1995 UMI. All rts. reserv.

00466673 89-38460
Real-Time UNIX Operating System : RX-UX 832

ABSTRACT: As microprocessor technology advances and the performance capabilities of microprocessor embedded computer systems improve,

requirements previously relevant only to large systems have become increasingly applicable to microprocessor embedded systems. To satisfy these requirements, NEC Corp. has developed a real-time UNIX operating system, called RX-UX 832, that runs on the manufacturer's original 32-bit microprocessors. RX-UX 832 has been developed in a building block approach, using the RTOS (real-time operating system) as the basic block. The new operating system is composed of 3 separate modules: 1. RTOS real-time kernel, 2. file-server, and 3. UNIX supervisor. To guarantee a real-time responsibility, several enhancements have been introduced, including a fixed priority task scheduling scheme, a contiguous block file system, and fault-tolerant functions. With RX-UX 832, system designers can use standard UNIX as a man-machine interface to construct fault-tolerant systems with sophisticated operability.

Mizuhashi, Yukiko; Teramoto, Masanori
Microprocessing & Microprogramming v27n1-5 PP: 533-538 Aug 1989 CODEN: MMICDT ISSN: 0165-6074 JRNL CODE: EUJ
DOC TYPE: Journal article LANGUAGE: English LENGTH: 6 Pages

22/7/2 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/INFORM(R)


00269266 85-09699
Universal Real-Time Kernel

ABSTRACT: A universal real-time kernel is proposed as a tool to facilitate real-time microprocessor system implementation. It consists of a library of kernel functions or basic system calls, which can be selected by programmers according to their particular needs. Major features of the universal real-time kernel include: 1. mechanisms for multitasking and interrupt handling, 2. functions that can be directly called by the programmer, delivered as modules, and added to the program at the time of its realization, 3. development environment independence, and 4. ease of extension, so that other system calls can be added without kernel modification. Basic kernel functions include operations on tasks, synchronization and communication, scheduling algorithms, real-time clock, and interrupt handling.

Maniecki, Marek
Microprocessing & Microprogramming v14n3,4 PP: 161-163 Oct/Nov 1984
CODEN: MMICDT ISSN: 0165-6074 JRNL CODE: EUJ
DOC TYPE: Journal article LANGUAGE: English LENGTH: 3 Pages

22/7/3 (Item 3 from file: 15)
DIALOG(R)File 15:ABI/INFORM(R)


00125170 80-19207
Meeting High Standards with the Extended Operating System

ABSTRACT: Bell Labs has recently developed a new general-purpose operating system, the Extended Operating System (EOS), for use with the 3A Auxiliary Processor (3A-AP); this combined software-hardware base can handle a broad variety of real-time, stored-program systems, achieving the high standards established by software for electronic switching systems. The system offers several advantages: 1. The cost of developing EOS can be spread among the systems using it and can be modified

for a new processor. 2. Programmers using EOS need consider only the details of the operating system since EOS itself handles all interfaces with hardware. 3. The system 's reliability makes it easier to develop new feature programs. The heart of the system , the EOS '' kernel ,'' schedules the use of the processor among the programs, transfers data among programs, schedules several low-level service routines to deal with interrupts , and handles system calls . Along with the kernel , EOS has several other packages, including a file system , terminal package, and maintenance package.

Elmendorf, Charles H.
Bell Laboratories Record v58n3 PP: 97-103 Mar 1980 CODEN: BLRCAB
ISSN: 0005-8564 JRNL CODE: BLR
DOC TYPE: Journal article LANGUAGE: English

22/7/4 (Item 1 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)


04640956
Streamlined microkernel
Microwave Systems: To intro operating system OS-9 Version 3.0

BY ALEXANDER WOLFE

Des Moines, Iowa - Aiming to deliver optimized systems software capable of running resource-limited embedded applications, Microware Systems Corp. will unveil today OS-9 Version 3.0.

The streamlined operating system features a preemptible kernel along with enhancements in realtime services. Leading the list is faster interrupt response, cut to 3 microseconds, versus 5.0 microseconds for the previous release of OS-9. Also featured is faster context switching , more efficient interprocess communications (IPC) mechanisms including the addition of binary semaphores, improved determinism, upgraded memory management and greater system call throughput.

Microware is targeting a broad range of embedded applications, but emphasizes what it calls 'hard' real-time apps where determinism and fast interrupt response are a necessity. These include telecommunications, process control and intelligent vehicle highway systems.

Most recently, the company has pressed for design wins in emerging multimedia applications. 'If you take a look at our support of MPEG and networking, we have the components for a drop-in solution for video-on-demand set-top applications,' said Steve Johnson, Microware's director of product marketing.

In terms of its construction, OS-9 V 3.0 consists of five functional layers. At the design's core is the kernel , providing basic OS services such as task and memory management, intertask communication and task synchronization. The I/O management layer controls the input/output subsystem. The subsequent layers consist of the file managers, device drivers and descriptors.

Along with the V 3.0 microkernel, Microware is releasing the Atomic OS-9 run-time microkernel. Atomic OS-9 excludes the development and debug functions found in V 3.0, which normally aren't required in runtime environments.


*FULL TEXT AVAILABLE IN FORMAT 9*
WORD COUNT: 367

Electronic Engineering Times October 4, 1993 p. 34
ISSN: 0192-1541

22/7/5 (Item 2 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)


03710563
Real-time o/s adds 'scalable' features
Ready Systems: Developing 'scalable architecture' real-time operating system (RTOS)

Ready Systems is developing, what it describes as a "scalable architecture" real-time operating system (rtos).

A minimum set of 15 operating system calls, described as a nanokernel, have been defined. At 4.5kbyte this occupies about one fifteenth the size of other microkernels developed to add real-time capabilities to Unix.

Users will be able to use the nanokernel for applications requiring the lowest level of capability with the highest performance. Additional calls and libraries of routines can be added up to and beyond the level of a full Posix (the IEEE standardised definition of Unix) operating system.

Both extremes could be supported simultaneously on an embedded system. This would allow systems to offer variable capabilities from "reflex" responses to high-priority interrupts while also performing computer and software-intensive operations such as networking or image processing.

Ready Systems has defined a library of calls which supersets those offered by its established rtos, VRTX32. By adding these to the nanokernel, compatibility for existing VRTX32 users will be provided under the name VRTXsa.

According to Bruce Gregory, president of Ready Systems, not only does VRTXsa offer emulation of VRTX32 plus 22 addition calls but it also runs 15% faster in typical applications.

The company will also offer a tool called Kernel -builder to allow users to emulate existing rtos using the nanokernel.

The nanokernel is being ported to the H8 microcontroller under a contract with Hitachi and is already running on a version of the MIPS R3000 microprocessor.

The first commercial products will be ports to the 68XXX microprocessors followed by ports to the X86 family.


WORD COUNT: 252

Electronics Times March 12, 1992 p. 9

22/7/6 (Item 3 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)


03694436
Solbourne Announces OS/MP Enhancements

Solbourne Computer, Inc., the leading supplier of SPARC-compliant symrnetric multiprocessing servers, today announced OS/MP 4.1A.2, an optional upgrade to its operating system which significantly improves

database performance. Offered at no cost to Solbourne customers under contract, OS/MP 4.1 A.2 is available immediately. The new release reflects Solbourne's enhancements for database performance made cooperatively with Oracle. Most symmetric enhancements in the new release are made completely in the OS/MP kernel . They include: o The types of I/O most common to database transactions - raw I/O and asynchronous I/O - were examined closely and refined. Operations common in database transactions such as timing operations were made more parallel. In addition, a lock was assigned to Itaw 1/0 that is not used by other kernel functions, reducing lock contention o Portions of the semop kernel semaphore code (a method of interprocess communication commonly used in database applications) have been rewritten to decrease lock contention between concurrent semaphore operations. The code was refined to allow more concurrency in database operations. o File system consistency check and interactive repair (fsck) was improved to speed disk integrity checks during booting without sacrificing accuracy. o Up to 1024 files in a database program can now be accessed concurrently - an increase of over four fold. o Context switching has been refined to use finer grain locks. The code executed between processes was modified with finer-grain locks, allowing multiple process switches to occur simultaneously. o System call changes were made to include multithreaded signal operations, a form of interprocess communication. These changes improve concurrency and scalability. o Improvements were made to the software TLB (translation lookaside buffer) coherency algorithm to minimize the number of inter-CPU interrupts sent.

Full text available on PTS New Product Announcements.

News Release February 24, 1992 p. 1

22/7/7 (Item 4 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)


02759631
Computer Harmony

Chorus Systems Inc., Beaverton, Ore., has introduced a micro-kernel -based operating system , called Chorus/MiX, that is designed to make optimal use of multiprocessing systems on networks. Chorus/MiX is compatible with Unix System V Release 3.2, and compatibility with Unix System V Release 4 is planned for next year. The vendor claims that because Chorux/MiX was designed from the start for multiprocessing computers on networks, it is more streamlined from versions of AT&T's Unix that have had multiprocessing and networking capabilities added. The kernel can support both fault -tolerant and real-time systems.


WORD COUNT: 89

CommunicationsWeek September 24, 1990 p. 48
ISSN: 0746-8121

22/7/8 (Item 5 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)

02329467
ULTIMATE AND SEQUOIA FORM JOINT MARKETING ALLIANCE

The Ultimate Corp. (NYSE: ULT) and Sequoia Systems, Inc., today announced a letter of intent to form a strategic business and marketing alliance. Under this exclusive worldwide agreement, Ultimate will remarket Sequoia's Pick-based line of fault tolerant computers. Under the letter of intent, Ultimate will invest approximately $5 million in the equity of the privately held Marlboro, Mass.-based Sequoia. In addition, each company will be represented on the other's board of directors by one new member, in each case, the company's chairman. In addition to fault tolerance and high availability, Sequoia hardware features modular growth -- so that customers buy only as much computing power as they need, and can expand over time -- as well as outstanding price/performance compared with competing products. Along with the hardware advantages, Ultimate sees another plus in Sequoia systems software configurations: Sequoia is a "native" Pick operating system machine built around a UNIX operating system kernel called TOPIX. This means that users can start with the exceptional application solutions based on Pick and then easily evolve to include UNIX-based programs. It also means the Pick users can take advantage of the powerful communications facilities built into Sequoia's UNIX- based systems. In order to consolidate its marketing focus for larger users, Ultimate will no longer offer its Tandem-based systems. The multi- year agreement announced today by Ultimate and Sequoia provides that Ultimate becomes Sequoia's exclusive Pick distributor worldwide.

Full text available on PTS New Product Announcements.

PR Newswire September 28, 1989 p. 1

22/7/9 (Item 6 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)


02129730
NEW REAL TIME UNIX COMPUTER INTRODUCED BY CONSOLIDATED COMPUTER SYSTEMS, INC.

The newest member of the OSI series700 family of 32-bit, single board, super microcomputers, all based on Motorala 680n0 processors, was announced at UNIXEPO today by Consolidated Computer System , Inc. (CCSI). The Ohio Scientific 720 runs under RTIX, an operating system that is fully compatible with UNIX's System V Interface Definition (SVID) at both the kernel and base extension levels. Not just a modification of UNIX, RTIX was written from scratch to incorporate real time capability within the kernel . The real time features of the RTIX kernel , such as NO WAIT system calls and REQUEST and EVENT queues, guarantee specific response times to external interrupts .

Full text available on PTS New Product Announcements.

News Release October 31, 1988 p. 1

22/7/10 (Item 7 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)


02097160

Real-Time Development Kit for 80386-Based PCs

A new software development package from Alcyon provides a fast path for creating a 32-bit PC-AT real-time UNIX-compatible platform, according to an announcement by Bill Allen, vice president of software. Specially priced for evaluation purposes, the REGULUS-386 Builder is a comprehensive development kit for 80386-based PC-AT and compaq compatible systems. It contains the complete REGULUS-386 operating system as well as a C compiler, assembler, linker, debugger, and support and upgrades for 90 days. REGULUS-386 allows the developer to take use of the full features of the 32-bit 80386. It combines all the features necessary for real- time operation with the familiar, rich development environment of a UNIX system . UNIX programmes written in C (or other high level languages) need simply be recompiled to run under REGULUS. ALL system calls and kernel features of UNIX System V are supported by REGULUS-386. Real-time operating system features provided include prioritized tasks, fast context switching efficient intertask communicaitons, contiguous files, and direct access to interrupts . Typical real-time latency (including interrupt disable, interrupt servicing, task scheduling and task switching ) on a 16 MHz 80386, with no wait-states, is 475 u seconds.

Full text available on PTS New Product Announcements.

News Release September 29, 1988 p. 1

22/7/11 (Item 8 from file: 16)
DIALOG(R)File 16:PTS Promt(TM)


01617516
SEQUOIA ANNOUNCES FIRST FAULT -TOLERANT PICK AND UNIX SUCCESS.

Sequoia Systems, Inc.today announced that it has completed a full implementation of Pick (TM) and UNIX (TM) on its Series 100 fault -tolerant on-line transaction processing (OLTP) system . The first demonstration of this achievement will be at the IDBMA's International Spectrum show in Las Vegas on March 23-25. In Sequoia's Pick implementation, the Series 100's operating system kernel replaces the standard Pick Monitor. While the Pick Virtual System has been ported using the normal macro expansion process, Virtual System monitor calls have been substituted with analogous kernel routines. The result is a true native Pick implementation running concurrently with the UNIX system . The Sequoia port represents the first fault -tolerant implementation of Pick and UNIX. Since the Sequoia Pick implementation is fully compatible with standard Pick environments, existinq Pick applications can be ported to the Series 100 without conversion. The Sequoia Pick implementation is also the first to provide availability of the Pick Open Architecture (OA). The Pick OA, the most advanced version of the Pick Relational DBMS, is fully compatible with the previous R83 version, offering improvements in performance, flexibility, and functionality.

Full text available on PTS New Product Announcements.

NEWS RELEASE March 23, 1987 p. 11
?